Title

Monte Carlo Methods for Sampling Protein Configurations

Author

Nilmeier, Jerome Paul

Publication Date

2008-09-18

Peer reviewed|Thesis/dissertation

Monte Carlo Methods for Sampling Protein Configurations

by

Jerome P. Nilmeier

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Biophysics

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

[redacted]

Acknowledgements

The material in Chapter 2 was previously published in the Journal of Computational and Theoretical Chemistry, and the material in Chapter 4 will be submitted for publication in nearly the same form.

I would like to thank the faculty and staff of the Graduate Group in Biophysics, particularly Julie Ransom and Rebecca Brown, for providing support, education, and inspiration. I would also like to thank the Tom Ferrin group at the UCSF Computer Graphics Laboratory, including Tom Goddard, Eric Pattersen, Conrad Huang, and especially Elaine Meng for providing support for creating figures and movies throughout my stay at UCSF. Most of the spectacular images that are presented here come from the Chimera package. I would also like to acknowledge the Pymol software package for providing a platform for creating images.

I would like to thank John Chodera for advising me to come to UCSF, and encouraging me to study statistical mechanics, even when it seemed too difficult. I also thank Huafeng Xu, Katarzyna Bernacki, Scott Shell, for discussions which have shaped my thinking about sampling methods in a lasting way.

I am grateful to Tack Kuntz and his group, including Terry Lang, David Sullivan, and Rob Rizzo for giving me a whirlwind introduction to the challenges of drug design.

Ken Dill has provided a foundation and environment for a delightful exploration of proteins and protein landscapes, for which I am deeply grateful. I am grateful to the entire group, including Vince Voelz, Banu Ozkan, Kingshuk Ghosh, Ilya Chorny, David Mobley, and many others for friendship and counsel.

I would like to thank Ivet Bahar and everyone involved with the Bioengineering and Bionformatics Summer Institute at the University of Pittsburgh and Carnegie Mellon. I found the experience to be deeply enriching, with a lasting impact on my later thesis work.

I thank Tanja Kortemme and the entire group for seeing new ways to effectively sample protein configurations. Colin Smith, Greg Friedland, and Dan Mandell have provided key insights into new sampling approaches by introducing new perspectives and fresh ideas.

Matt Jacobson has provided a wealth of scientific resources to draw from, and I am grateful for his support. While my interest has always gravitated towards the abstract, he has dutifully kept me mindful of the extraordinary chemical detail of biochemical systems. I thank Ben Sellers, Eli Groban, Arjun Narayanan, and all of the Jacobson group for their good companionship. Chris McClendon has been both a great friend and a courageous early adopter of fragile code and new theories. Pradipta Bandyopadhyay also provided key insights into prototype codes that are presented here, for which I am also grateful.

The Protein Local Optimization Program (PLOP) and Schrödinger development community have provided access to excellent software and source code, without which this work would not be possible. I thank Matt again for access to this resource, and the support provided in understanding the code fundamentals. He has always managed to provide the right level of support, commensurate with my needs and skill level. I also thank Michael Shirts, Woody Sherman, Ramy Farid, and Ken Borrelli for enriching conversations, incisive advice, and inclusion into a community of which I am grateful and proud to have been a member.

Evangelos Coutsias has provided important mentorship for me at many stages during my development, and I am thankful for his instruction and guidance. He has also provided key elements of the loop closure code that I was able to interface to the software. His commitment to providing this functionality allowed me to explore other aspects of the project in more depth.

I thank Teresa Head-Gordon and her group for granting a forum for expressing new ideas, even in their nascent forms. Nick Fawzi was extremely helpful in sustaining this relationship, while also being a good friend. I also thank David Chandler, Phil Geissler, and Gavin Crooks for expressing the ideals of statistical mechanics that I have always aspired to, and for providing access to the annual mini-Statmech Meetings, from which I have learned so much. I also thank my undergraduate mentors Harvey Blanch and John Prausnitz, for getting me excited about theoretical work related to biochemistry. I thank Prof. Prausnitz in particular seeing a scientist in me when I was unable to see it in myself, and for treating me like a scientist even in times when I wasn't always behaving as one. I am grateful for his kind mentorship and thoughtful conversation during my formative years.

Finally, I want to thank everyone in my family and Jennifer's family for encouragement and support. In particular, I want to thank my Mother, my Father, and Jill, who never failed to show their faith in me, even in difficult times.

to

Jennifer,

Sofia, and David

for a home full of good food, laughter, and joyful chaos

Abstract

A multiscale, modular approach to protein sampling with novel Monte Carlo algorithms is is presented. The systems studied use an all atom forcefield with a Generalized Born implicit solvation model. The multiscale approach addresses 3 degrees of freedom: 1) the solvation terms, 2) the sidechain degrees of freedom, and 3) the backbone degrees of freedom. The goal of the work is to identify the special design issues surrounding these degrees of freedom, and create an overall sampling approach that optimizes all of these, while generating coherent trajectories that obey detailed balance. This design is expected to sample challenging, highly constrained systems that may be exceedingly difficult using standard molecular dynamics methods. The work presents present developments with regard to algorithmic approaches, design features, and applications to protein systems. Future directions in these areas are also discussed.

TABLE OF CONTENTS

Chapter 1

Monte Carlo Methodologies

Introduction

Protein Configurations

Proteins are long, unbranched chains of amino acids that self assemble in a process known as folding to form secondary and tertiary structures. In general, for a known sequence of amino acids, there is one distinct protein structure. These folded structures act in nearly every aspect of cellular function, by facilitating chemical reactions and providing physical structure to the cellular machinery. Understanding the structure of a protein at a molecular level is key element of structural biology and of biophysics.

A vast field of study is dedicated to obtaining the crystal structure of the protein experimentally. There is also considerable energy dedicated to predicting protein structures computationally. Currently, protein structures are predicted using either bioinformatics or physics based (*ab initio*) methods. Bioinformatics approaches often rely on some knowledge about the sequence and structure. There are many portions of a protein for which bionformatics methods fail, and this requires the use of more predictive methods. The *ab initio* prediction field seeks to use strictly physical methods to predict structure.

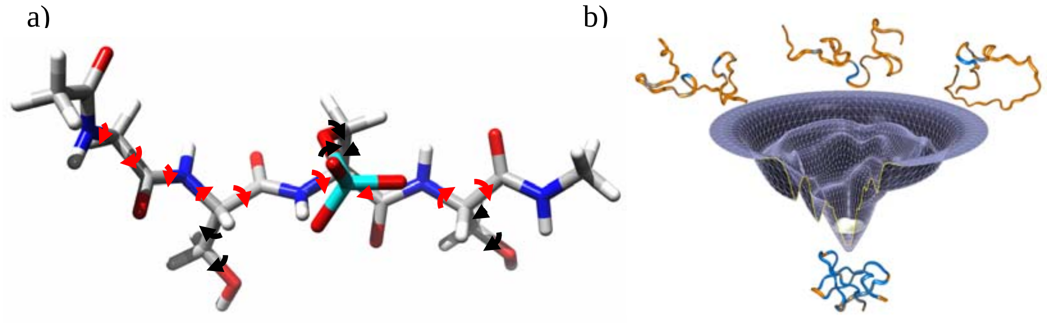


Figure 1.1 – a) Example of short polypeptide with backbone and sidechain dihedral degrees of freedom. Backbone dihedrals (ϕ) are shown in red, and sidechain dihedrals (χ) are shown in black. b) The energy landscape of a polypeptide as it folds into a native structure (Figure 1b courtesy of K.Dill)

Sampling Protein Configurations

The topic of structural biology and computational structure modeling is vast, and only a brief background is presented. The goal here is to present many of the essential mathematical ideas that will form the basis of the remainder of the work.

Due to the extraordinary geometric complexity of proteins, there exists a large body of work dedicated to exploring their structures. A key abstraction that is made

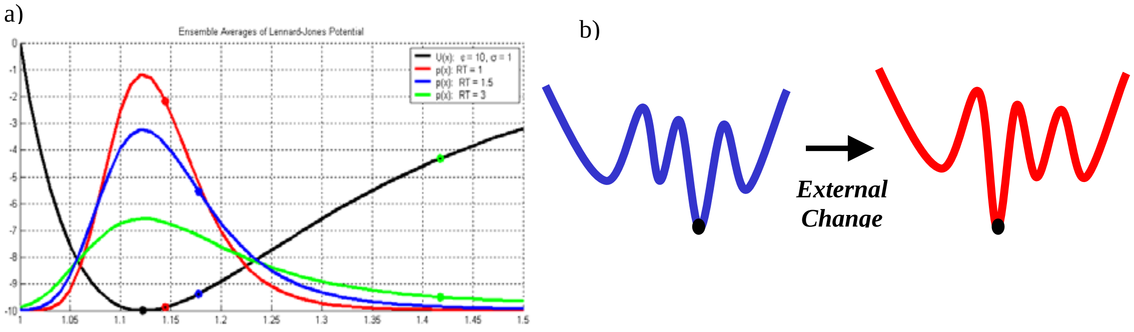


Figure 1.2 – Motivation for thermal sampling of protein structures. a) Differences between mean energies and the minimum energy. The black trace is a potential energy well with depth $\varepsilon=10RT_0$. Distributions and means are shown at T_0 , $1.5T_0$ and $3T_0$. a) Schematic of the global landscape of a protein in different environments.

when considering the configurations of a protein is to consider the energetics as a high dimensional landscape. A mainstay of the field is the hypothesis that the protein landscape is funnel-like [1-3]. Figure 1.1 shows a small peptide, as well as a landscape description of protein folding. The funnel hypothesis suggests that the landscape is deeper than it is wide, and that the lowest energy states are stabilized by the depth of the basins. Figure 1.2 shows how basin depth affects the mean value of an observed configuration. If the well is sufficiently deep, optimization methods may be utilized to locate local minima. In general, however, it is desirable to locate many local minima, as well as to estimate the population of each of these states. One motivation for this is presented in Figure 1.2b, whereby adjacent local minima may play a role in stabilizing

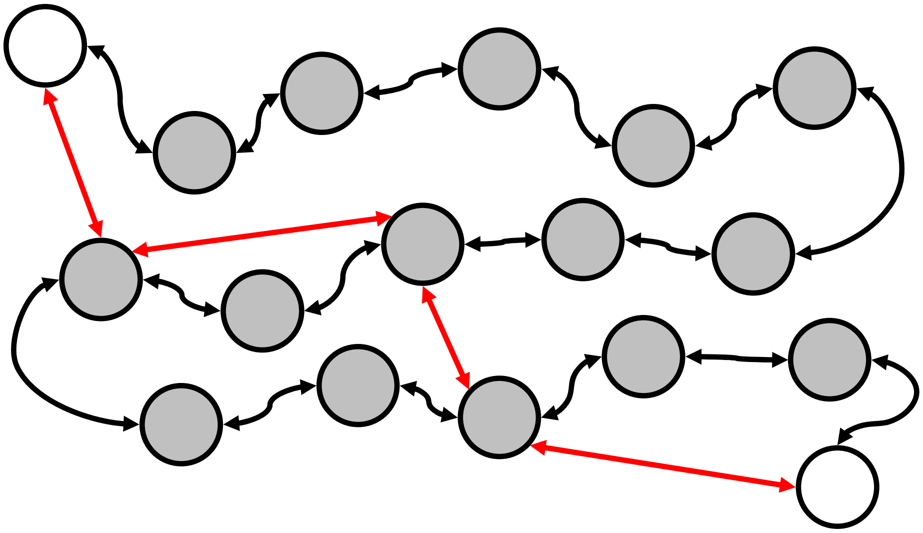


Figure 1.3 – Trajectories through configuration space with Monte Carlo depends on the trial move set. The circles represent positions in state space, or locations along an energy landscape. The white circles represent 'interesting' configurations. The black arrows represent the connectivity of states using a dynamical propagator, such that each state is connected through space and time. The red arrows represent a new connectivity of states due to a Monte Carlo strategy. An ideal Monte Carlo scheme will sample a broader region of configurations more efficiently.

interactions with undetermined changes in conditions, such as the presence of a substrate. To predict macrostate populations correctly requires either molecular dynamics methods or Monte Carlo methods. Molecular dynamics methods are by far the most widely used, but can often suffer from being unable to sample space completely. Monte Carlo methods can often give very good sampling, if the trial move set is constructed correctly.

Figure 1.3 shows how a cleverly constructed trial move set can efficiently sample across an energy landscape to locate interesting configurations. The main focus of this work is to present developments of Monte Carlo methodologies to facilitate an efficient sampling of protein landscapes.

Monte Carlo: Theoretical Foundations

Monte Carlo Integration

Monte Carlo[4] is a well known method for simulating molecular systems, and the foundations are well understood[5-7]. As has been described previously, it is also a method for numerically evaluating an integral. It is common to view the definition of the integral as a prescription for sampling in the physical system of interest. More specifically, if we are able to define an integral such that

Equation 1

$$Q = \int d\mathbf{X} f(\mathbf{X})$$

where **X** is a vector with dimension *N*, such that

Equation 2

$$\mathbf{X} = [x_1, x_2, \ldots x_N]^T$$

and the differential is expressed compactly as

Equation 3

$$d\mathbf{X} = dx_1 dx_2 ... dx_N = \prod_{i=1}^{N} dx_i$$

This integral is often referred to as the *partition function*, or *configuration integral*. To evaluate Eq. 1 numerically, the standard approach would be to discretize the space into small volume elements $\delta\mathbf{X}_i$, which can be of varying size, depending on the position in space, and evaluate the following sum over the entire volume Ω:

Equation 4

$$Q = \sum_{i \in \Omega} \delta\mathbf{X}_i f(\mathbf{X}_i)$$

As one might expect, the difficulty in evaluating this integral for high dimensional systems is in constructing the volume elements correctly, such that the regions of space contributing to the integral are adequately sampled, and that the numerical error in the discretization of the space is minimized. As the number of dimensions increase, a different methodology becomes ultimately a more efficient approach. The approach is simply to select a point $\boldsymbol{\xi}_i$ in the N space from a uniform distribution, and compute the following sum:

Equation 5

$$Q = \frac{\Omega}{N_T} \sum_{i=1}^{N_T} f(\boldsymbol{\xi}_i)$$

where N_T is the total number of trials. This method proves to be more efficient for high dimensional systems than constructing N_T bins and evaluating Eq. 4. While the body of this work does not incorporate Monte Carlo integration, it is presented as a motivation for using stochastic methods to sample very large configuration spaces.

Importance Sampling

As the size of the configuration space becomes larger, the density of the space for which the function $f(\mathbf{X})$ contributes substantially to the integral becomes smaller and smaller. It therefore becomes of interest to sample the volume elements with more 'importance' more frequently. If we identify the function $f(\mathbf{X})$ as a (normalized) probability distribution $p(\mathbf{X})/Q$, we can see more easily that the high probability regions of space will contribute more to the integral than the low probability portions. Moreover, if we are interested in some observable property of the system:

Equation 6

$$<O>=1/Q\int d\mathbf{X}O(\mathbf{X})p(\mathbf{X})$$

where $p(\mathbf{X})$ is an unnormalized probability function associated with a physical model through a Boltzmann factor of some energy function:

Equation 7

$$p(\mathbf{X})=e^{-\beta U(\mathbf{X})}$$

where β=1/k_BT is the inverse temperature T times the Boltzmann constant k_B. If we are able to visit regions of configuration space with the same probability as is given by Eq. 7, then we can simply evaluate Eq. 6 as:

Equation 8

$$< O >= \frac{1}{N_T} \sum_{i=1}^{N_T} O(\mathbf{X}_i)$$

where N_T is again the number of trials. It is important to notice that the evaluation of Eq. 8 does not require knowledge of the normalization constant Q. This method will be even more efficient than the Monte Carlo integration technique, because the sites visited are only the high probability (or low energy) states of the system, which typically comprise a much smaller subspace of the entire system of interest. The goal, then, is to devise a strategy that will allow us to visit regions of configuration space with the correct probability.

Master Equation and Balance Requirements

The master equation model[8] is a common starting point in studying broader class of statistical mechanical problems, and it can be helpful in understanding the ideas leading to the more familiar Metropolis criterion that is widely used in Monte Carlo techniques. To begin, an index is assigned to every coordinate state:

Equation 9

$$p_i = p(\mathbf{X}_i)$$

There is no particular restriction with regard to the use of continuous coordinates here. The discretization is made only for convenience in notation. The probabilities need not be normalized, and this will become a general feature of the sampling approach.

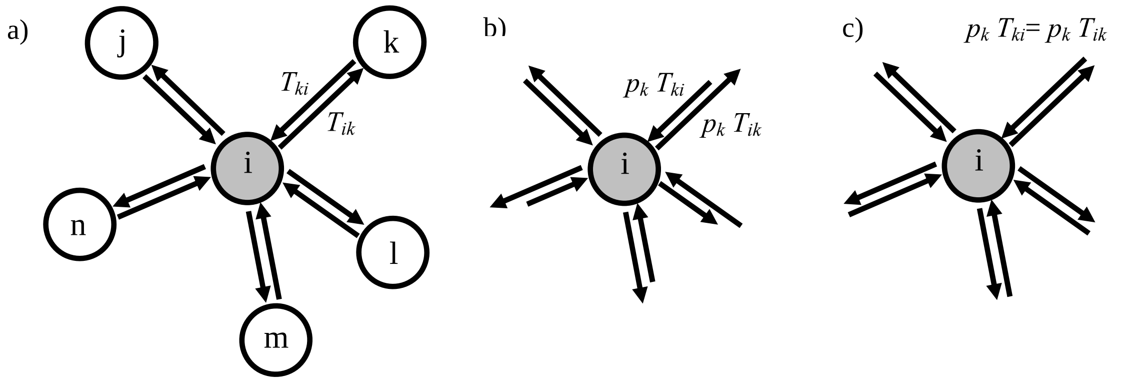


Figure 1.4 – Examples of bookkeepping for probability flowrates. a) Flow of probability through state *i* is shown. For simplicity, only states *j-n* flow to state *i*. (For figures b and c, arrows have length commensurate with the product of the probability times the transition probability, or 'flowrate of probability' b) Balance. The sum of the incoming flowrates is equal to the sum of the outgoing flowrates. c) Detailed balance. All flowrates at each node are equal.

A simple statement of the conservation of probability over the entire configuration space at all times gives the Master Equation:

Equation 10

$$\frac{dp_i}{dt} = \sum_{j \neq i} p_j T_{ji} - \sum_{j \neq i} p_i T_{ij}$$

which is simply a first order kinetic model for the flow between states of a system. Here, T_{ij} is defined to be the *transition probability* from state *i* to state *j*, (See Figure 1.4). We wish to devise some propagation strategy that will ensure that the final state of the system has a stationary probability distribution over all states. We simply set the left hand side of Eq. 10 to zero and obtain this condition, which is the condition of *balance*[9]:

Equation 11

$$\sum_{j \neq i} p_i T_{ij} = \sum_{j \neq i} p_j T_{ji}$$

This is the most general requirement for a Monte Carlo propagation strategy that will ensure that a stationary distribution is obtained (see figure 1.4b). For this work, however, the stricter condition of *detailed balance* is enforced for all pairs of states *i* and *j*:

Equation 12

$$p_i T_{ij} = p_j T_{ji}$$

this stricter condition ensures that Eq. 11 is satisfied. This condition is illustrated in Figure 1.4c. In practice, this condition is easier to enforce, and is used more widely. This propagation strategy will ensure that sites are visited with a frequency commensurate with the probability of occupying that site. A sequence of states visited according to this prescription is known as a *Markov Chain* of states, and a propagation scheme that follows this strategy is known as a *Markov Chain Monte Carlo Method*.

The Metropolis-Hastings Acceptance Criterion

Since the probabilities of each state are known, Eq. 12 is a prescription for the transition probability. Hastings[10] has provided a formalism for understanding the Markov chain method that is a generalization of the earliest developments of Rosenbluth[4,11,12], and the foundations for the biased sampling[13] methods that are widely used, and form the basis of many of the techniques which are presented in the present work[14-16].

The transition probability is defined as the product of two probabilities:

Equation 13

$$T_{ij} = \alpha_{ij} acc_{ij}$$

where α_{ij} is the *selection probability*, and acc_{ij} is the *acceptance* probability, or *acceptance rule*. The selection probability is the designed feature of the propagation strategy. It is the probability of selecting coordinate *j* from coordinate *i*. The most common design used here is some uniform selection probability (adding a uniform deviate to the coordinate state is a common choice). As we shall see, the choice of uniform deviates has the convenient property that $\alpha_{ij} = \alpha_{ji}$ for all states. Of course, this is not the only choice, and a clever construction of these distributions forms the basis of many of the methods introduced throughout this work.

Given the selection probability, there remains only to solve for the acceptance probabilities, which form the basis of the propagation rule. Combining Eqs. 12 and 13 gives:

Equation 14

$$\frac{acc_{ij}}{acc_{ji}} = \frac{\alpha_{ji} p_j}{\alpha_{ij} p_i}$$

this ratio of forward and reverse acceptance probabilities defines the acceptance rule in terms of known quantities. For any single propagation step, however, we would like to know what the acceptance rule is. There are two commonly used functions that satisfy Eq.14. The less widely known function is the Barker acceptance rule:

Equation 15

$$acc_{ij} = \frac{\alpha_{ji} p_j}{\alpha_{ij} p_i + \alpha_{ji} p_j}$$

Eq. 15 is evaluated by choosing a random number uniformly distributed over [0,1], and comparing to the value computed. If the random number is less than the number computed, then the move is accepted. If not, then the move is rejected. The more familiar Metropolis acceptance rule is:

Equation 16

$$acc_{ij} = \min\left(1, \frac{\alpha_{ji} p_j}{\alpha_{ij} p_i}\right)$$

where the move is always accepted if the ratio in the argument is greater than 1. If the ratio is less than one, a random uniform number over [0,1] is selected and compared to the ratio. Again, if the random number is less than the ratio, the move is accepted. A seemingly endless variety of functions should exist that satisfy Eq. 14, but, in practice, nearly all acceptance rules use the Metropolis criterion some exceptional cases using the Barker criterion. All of the modified acceptance criteria presented in this work can be derived from Eq. 14.

Hybrid Monte Carlo

An important Markov Chain Monte Carlo algorithm of particular interest is the Hybrid Monte Carlo[17] approach of Duane and Kennedy. While it is not a fundamental theory of Monte Carlo sampling, it is of particular interest in relation to some of the algorithms developed for this work, and is presented here as background.

The basic idea behind Hybrid Monte Carlo is to incorporate Molecular Dynamics moves in a Monte Carlo scheme in a way that obeys detailed balance. This can prove to be a powerful method in complex systems, since gradient information can be incorporated in the trial move set. It can be thought of as a way of generating a more "natural" trial move in local space. It does not, however, sample large regions of space, since the trial moves only randomize the initial velocities, and not the positions. In later sections, we shall introduce a companion algorithm to HMC which may prove to be useful in improving ergodicity of HMC algorithms.

Procedural Details of Hybrid Monte Carlo

The basic steps in a single Hybrid Monte Carlo move are as follows:

1) Select momenta from a Gaussian distribution
2) Use a molecular dynamics algorithm to propagate for a number of steps
3) Accept the move using the Metropolis criterion applied to the change in the Hamiltonian

The procedure for a single Hybrid Monte Carlo step is to first select momenta from a Gaussian distribution of velocities:

Equation 17

$$p_G(\boldsymbol{\pi}) = \exp\left(-\frac{\beta}{2}\mathbf{M}^{-1}\boldsymbol{\pi}\cdot\boldsymbol{\pi}\right)$$

where **π** = **Mv** is the momentum vector, **M** is a matrix containing the masses of each of the atoms, such that $\mathbf{M}_{ii} = m_i$, and **v** is the vector of velocities. To show how the velocities are generated in practice, we express Eq. 17 as a product of Gaussian distributions:

Equation 18

$$\begin{aligned} p_G(\boldsymbol{\pi}) &= \prod_{i=1}^{N} \exp\left(-\frac{\beta m_i v_{x,i}^2}{2}\right)\exp\left(-\frac{\beta m_i v_{y,i}^2}{2}\right)\exp\left(-\frac{\beta m_i v_{z,i}^2}{2}\right) \\ &= \prod_{i=1}^{N} \exp\left(-\frac{\beta m_i \mid \mathbf{v}_i \mid^2}{2}\right) \end{aligned}$$

The process for selecting these velocities is, then, to assign a random velocity of unit magnitude for each atom, and select the magnitude of the velocity from a gaussian distribution with standard deviation σ_i given by:

Equation 19

$$\sigma^2{}_i = 1/\beta m_i$$

Gaussian variates are generated by selecting from a uniform distribution and applying the Box-Muller transform[18].

After selecting velocities, a short molecular dynamics trajectory is generated. The Verlet operator L_V is applied to the initial conditions can be expressed as:

Equation 20

$$(\mathbf{q}', \boldsymbol{\pi}') = L_V(\mathbf{q}, \boldsymbol{\pi} = \boldsymbol{\xi})$$

where **q'** and **π'** are generated by applying the Verlet algorithm to the coordinates **q** and **ξ**, where **ξ** is the vector of momenta generated as described by Eqs. 18 and 19. Here, the subscript *V* refers to the potential that drives the Verlet propagation strategy.

The Verlet Leapfrog propagator is defined in the usual way. The initial half-step:

Equation 21

$$\boldsymbol{\pi}_1 = \boldsymbol{\xi} - \nabla V(\mathbf{q})\delta\tau / 2$$

is followed by $S = \tau/\delta\tau$ steps, indexed by t, in position space and S -1 steps in momentum space:

Equation 22

$$\begin{aligned}\boldsymbol{\pi}_t &= \boldsymbol{\pi}_{t-1} - \nabla V(\mathbf{q}_{t-1})\delta\tau, \ (t > 1) \\ \mathbf{q}_t &= \mathbf{q}_{t-1} + \mathbf{M}^{-1}\boldsymbol{\pi}_t \delta\tau\end{aligned}$$

and the final half step:

Equation 23

$$\boldsymbol{\pi}_S = \boldsymbol{\pi}_{S-1} - \nabla V(\mathbf{q}_S)\delta\tau / 2$$

and the final velocities and positions are assigned **(q',π')** = (**q**$_S$, **π**$_S$). The Verlet strategy has two important properties to notice. The first is that the algorithm deterministic and time reversible, which means that the probability of arriving at a state is given by a Dirac function:

Equation 24

$$\alpha_V(\mathbf{q}, \boldsymbol{\pi} \to \mathbf{q}', \boldsymbol{\pi}') = \delta(\mathbf{q}' - \mathbf{q}_S, \boldsymbol{\pi}' - \boldsymbol{\pi}_S)$$

which is true for a deterministic propagation of the coordinates. The time reversibility condition simply states that the trajectory run in reverse will trace the same path that the forward trajectory traces. In terms of selection probabilities, this can be expressed as:

Equation 25

$$\alpha_V(\mathbf{q}, \boldsymbol{\pi} \to \mathbf{q}', \boldsymbol{\pi}') = \alpha_V(\mathbf{q}', -\boldsymbol{\pi}' \to \mathbf{q}, -\boldsymbol{\pi})$$

The second important feature of the Verlet algorithm is that the Hamiltonian is preserved to order of $\delta\tau^2$. The Hamiltonian *H(***π,q***)* is given by the sum of the potential *V*(**q**) and kinetic *K*(**π**) energies:

Equation 26

$$H(\boldsymbol{\pi}, \mathbf{q}) = V(\mathbf{q}) + K(\boldsymbol{\pi})$$
$$K(\boldsymbol{\pi}) = 1/2(\mathbf{M}^{-1}\boldsymbol{\pi}) \cdot \boldsymbol{\pi}$$

Expressing these terms as Boltzmann factors gives:

Equation 27

$$\begin{aligned} p_V(\mathbf{q}) &= \exp\left(-\beta V(\mathbf{q})\right) \\ p_K(\boldsymbol{\pi}) &= \exp\left(-\beta K(\boldsymbol{\pi})\right) = p_G(\boldsymbol{\pi}) \end{aligned}$$

and observe that the Boltzmann factor of the kinetic energy is the Gaussian distribution given by Eq. 17. The Boltzmann factor of the Hamiltonian is given by:

Equation 28

$$\begin{aligned} p_{H,V}(\mathbf{q},\boldsymbol{\pi}) &= p_V(\mathbf{q})\,p_K(\boldsymbol{\pi}) \\ &= p_V(\mathbf{q})\,p_G(\boldsymbol{\pi}) \end{aligned}$$

where the notation $p_{H,V}$ also conveys the potential being used to propagate the coordinates. It should be noted that the only requirement for a proper potential here is that there are associated analytical gradients that can be computed in order to conserve the Hamiltonian of the system. This property is true of potentials that may or may not be representative of the complete physical description of the system. With this background and notation in hand, the acceptance criterion is defined simply as:

Equation 29

$$\left.\frac{acc(\mathbf{q}\rightarrow\mathbf{q}')}{acc(\mathbf{q}'\rightarrow\mathbf{q})}\right|_{HMC} = \frac{p_{H,V}(\mathbf{q}',\boldsymbol{\pi}')}{p_{H,V}(\mathbf{q},\boldsymbol{\pi})} = \frac{\exp\left[-\beta H(\mathbf{q}',\boldsymbol{\pi}')\right]}{\exp\left[-\beta H(\mathbf{q},\boldsymbol{\pi})\right]}$$

where the convention of expressing the ratio of acceptance probabilities, as in ratio of acceptance probabilities, as in Eq. 14, is adopted. Since the Hamiltonian is conserved, the acceptance ratio for these processes is nearly unity, and the size of the timestep can be adjusted to ensure this to within a desired range.

Proof of Detailed Balance

Rather than solving for the acceptance probabilities, detailed balance will be demonstrated. Eq. 29 can be expressed using the terms of Eq. 28:

Equation 30

$$\frac{acc(\mathbf{q},\boldsymbol{\pi} \to \mathbf{q}',\boldsymbol{\pi}')}{acc(\mathbf{q}',\boldsymbol{\pi}' \to \mathbf{q},\boldsymbol{\pi})} = \frac{p_{H,V}(\mathbf{q}',\boldsymbol{\pi}')}{p_{H,V}(\mathbf{q},\boldsymbol{\pi})} = \frac{p_V(\mathbf{q}')p_G(\boldsymbol{\pi}')}{p_V(\mathbf{q})p_G(\boldsymbol{\pi})}$$

Rearranging Eq. 30 gives:

Equation 31

$$p_\mathrm{V}(\mathbf{q})p_\mathrm{G}(\boldsymbol{\pi})\mathrm{acc}(\mathbf{q},\boldsymbol{\pi} \to \mathbf{q}',\boldsymbol{\pi}') = p_\mathrm{V}(\mathbf{q}')p_\mathrm{G}(\textbf{-}\boldsymbol{\pi}')\mathrm{acc}(\mathbf{q}',-\boldsymbol{\pi}' \to \mathbf{q},-\boldsymbol{\pi})$$

where we notice that $p_\mathrm{G}(\boldsymbol{\pi}') = p_\mathrm{G}(\textbf{-}\boldsymbol{\pi}')$, due to the symmetry of the Gaussian distribution. Multiplying both sides by the appropriate expressions in Eq. 25:

Equation 32

$$\begin{aligned} & p_\mathrm{V}(\mathbf{q})p_\mathrm{G}(\boldsymbol{\pi})\alpha_V(\mathbf{q},\boldsymbol{\pi} \to \mathbf{q}',\boldsymbol{\pi}')\mathrm{acc}(\mathbf{q},\boldsymbol{\pi} \to \mathbf{q}',\boldsymbol{\pi}') \\ &= p_\mathrm{V}(\mathbf{q}')p_\mathrm{G}(\boldsymbol{\pi}')\alpha_V(\mathbf{q}',\textbf{-}\boldsymbol{\pi}' \to \mathbf{q},\textbf{-}\boldsymbol{\pi})\mathrm{acc}(\mathbf{q}',-\boldsymbol{\pi}' \to \mathbf{q},-\boldsymbol{\pi}) \end{aligned}$$

Since the momenta are sampled from a distribution, it can be integrated out. To do this, we multiply both sides by $d\boldsymbol{\pi}d\boldsymbol{\pi}'$ and integrate. The left hand side of Eq.32 is:

Equation 33

$$p_\mathrm{V}(\mathbf{q})\int d\boldsymbol{\pi}d\boldsymbol{\pi}'p_\mathrm{G}(\boldsymbol{\pi})\alpha_H(\mathbf{q},\boldsymbol{\pi} \to \mathbf{q}',\boldsymbol{\pi}')\mathrm{acc}(\mathbf{q},\boldsymbol{\pi} \to \mathbf{q}',\boldsymbol{\pi}') = p_\mathrm{V}(\mathbf{q})\mathrm{T}(\mathbf{q} \to \mathbf{q}')$$

where $\mathrm{T}(\mathbf{q} \to \mathbf{q}')$ is probability of transitioning to state **q'** from state **q**. The right hand side of Eq. 32 is computed similarly as:

Equation 34

$$p_\mathrm{V}(\mathbf{q})\int d(\textbf{-}\boldsymbol{\pi})d(\textbf{-}\boldsymbol{\pi}')p_\mathrm{G}(\textbf{-}\boldsymbol{\pi})\alpha_H(\mathbf{q}',\textbf{-}\boldsymbol{\pi}' \to \mathbf{q},-\boldsymbol{\pi})\mathrm{acc}(\mathbf{q}',\textbf{-}\boldsymbol{\pi}' \to \mathbf{q},\textbf{-}\boldsymbol{\pi}) = p_\mathrm{V}(\mathbf{q}')\mathrm{T}(\mathbf{q}' \to \mathbf{q})$$

where it is noted that $d\boldsymbol{\pi}d\boldsymbol{\pi}' = d(-\boldsymbol{\pi})d(-\boldsymbol{\pi}')$. The results of Eqs. 33 and 34 demonstrate that detailed balance is obeyed with regard to the coordinate:

Equation 35

$$p_{\mathrm{V}}(\mathbf{q})\mathrm{T}(\mathbf{q} \to \mathbf{q}) = p_{\mathrm{V}}(\mathbf{q}')\mathrm{T}(\mathbf{q}' \to \mathbf{q})$$

It should be noted that the choice of distribution for the initial momenta are not arbitrary. The main requirement is that the distribution be symmetric, such that the identity used to transform Eq. 30 to Eq. 31 can be invoked. The acceptance criterion from which the proof has been developed requires that the distributions come from the specific distribution defined in Eq. 27. It is of course, possible, to select from a distribution other than this distribution, but this would also require a modified acceptance probability.

Hybrid Monte Carlo algorithms are, in general, a nice complement to standard MCMC methods, since the acceptance probabilities are very high. The use of gradient information facilitates the generation of low energy trial states. The incorporation of randomized momenta also provide for trial moves to emerge from steep enthalpic basins to a more entropically favored state that can often help to alleviate pathological kinetic trapping using standard Monte Carlo moves. In general, Hybrid Monte Carlo moves in the context of some of the algorithmic developments represents an interesting future direction for this work.

Configuration Integral for a Protein

Having established that Monte Carlo sampling is a means for evaluating integrals of the type given by Eqs. 6-8, we can proceed to assign a more physical description of the model, and introduce the form of the configuration integral which will be used in the remaining chapters. The partition function begins as:

Equation 36

$$Q = \int d\{\mathbf{R}\} e^{-\beta[U(\mathbf{R})]}$$

where **R** is the set of all coordinates for both protein and solvent, and the potential $U(\mathbf{R})$ is the pairwise (molecular mechanical) potential describing all interactions of protein and solvent. We can separate the contributions of each and express the partition function as:

Equation 37

$$Q = \int d\{\mathbf{R}^{<P>}\} d\{\mathbf{R}^{<W>}\} e^{-\beta[U_P(\mathbf{R}^{<P>}) + U_W(\mathbf{R}^{<W>}) + U_{PW}(\mathbf{R}^{<P>}, \mathbf{R}^{<W>})]}$$

where $\mathbf{R}^{<P>}$ is the set of all (Cartesian) protein coordinates, $\mathbf{R}^{<W>}$ is the set of all solvent coordinates. Here, the potentials U_P and U_W indicate the portions of the potential that depend only on the coordinates of the protein and the solvent, respectively. The last potential depends on the coordinates of both protein and solvent. Following the traditional formulation of implicit solvation[19], it is possible to assert a new potential $<U_{PW}(\mathbf{R}^{<P>})>$, which is a function of the protein coordinates only. We can then integrate out the coordinates of the solvent:

Equation 38

$$Q = \int d\{\mathbf{R}^{<P>}\} e^{-\beta[U_P(\mathbf{R}^{<P>}) + <U_{PW}(\mathbf{R}^{<P>})>]} \int e^{-\beta U_W(\mathbf{R}^{<W>})} d\{\mathbf{R}^{<W>}\}$$

Equation 39

$$= C\int d\{\mathbf{R}^{<P>}\}e^{-\beta[U_P(\mathbf{R}^{<P>})+<U_{PW}(\mathbf{R}^{<P>})>]} = C\int d\{\mathbf{R}^{<P>}\}e^{-\beta[A(\mathbf{R}^{<P>})]}$$

where $A(\mathbf{R}^{<P>}) = U_P(\mathbf{R}^{<P>}) + <U_{PW}(\mathbf{R}^{<P>})>$ is a new potential that contains the contribution of the protein interactions and the average contributions of the solvent molecules. This is likened to a free energy, since it depends on average properties. For our purposes, however, we can view the new potential as defining our new partition function. For this work, $<U_{PW}(\mathbf{R}^{<P>})>$ is estimated using the Surface Generalized Born Model (SGB)[20]. Many of the sampling approaches introduced in Chapter 2 rely on the notion that the solvent model requires more CPU time to evalutate than the remaining elements of the forcefield.

Following Deem's derivation[21], we can perform a change of coordinates to a local coordinate system:

Equation 40

$$\mathbf{r}_1 = \mathbf{R}^{<P>}{}_1$$
$$\mathbf{r}_i = \mathbf{R}^{<P>}{}_i - \mathbf{R}^{<P>}{}_{i-1}$$
$$d\mathbf{r} = d\mathbf{R}$$

The first equation fixes the rotational and translational degrees of freedom. This representation is sometimes referred to as the *bond vector representation*, originally described by Flory and later by Scheraga. The partition function can further be separated into:

Equation 41

$$Q = C\int d\{\mathbf{r}\}e^{-\beta[A(\mathbf{R}^{<P>})]}$$

We can now express coordinates in spherical coordinates of the *i*th coordinate system:

Equation 42

$$d\mathbf{r}_i = dx_i dy_i dz_i = l_i^{\,2} \sin(\theta_{i-1}) dl_i d\theta_{i-1} d\Phi_{i-1}$$
$$d\mathbf{r} = \prod_{i\in R} l_i^{\,2} \sin(\theta_{i-1}) dl_i d\theta_{i-1} d\Phi_{i-1}$$

where θ and Φ are the bond and dihedral angles, respectively. Expressing the partition function in terms of these coordinate and invoking the constraint that the bond angles and lengths are preserved yields:

Equation 43

$$Q = C\int d\mathbf{r}\prod_{i\in R} l_i^2 \sin(\theta_{i-1}) dl_i d\theta_{i-1} d\phi_{i-1} e^{-\beta[A(\mathbf{R}^{<P>})]}\{\delta^3(\boldsymbol{\theta}^{<N>} - \boldsymbol{\theta}_{i,0}{}^{<N>})\}\{\delta^3(\mathbf{l}^{<N>} - \mathbf{l}_{i,0}{}^{<N>})\}$$
$$= C'\int d\mathbf{\Phi} e^{-\beta[A(\mathbf{R}^{<P>})]}$$

The important thing to notice about the above transformation is that there is no effective Jacobian term in the integrand as a result of transforming into dihedral coordinates. The configurational integral can now be expressed in terms of the backbone dihedrals $\boldsymbol{\phi}$, and the sidechain coordinates $\boldsymbol{\chi}$:

Equation 44

$$Q = C''\int d\boldsymbol{\varphi} d\boldsymbol{\chi} e^{-\beta[A(\mathbf{R}^{<P>})]}$$

where the sidechain dihedrals are encompass all rotatable bonds along the sidechain branches of a peptide, and the backbone dihedrals cover all rotatable bonds of a backbone (ϕ,ψ,ω). For most of the work, however, the ω angles are held fixed at the native states. The Dirac transformation for this constraint is straightforward, and not shown here. Figure 1.4 shows an example of a small peptide and the associated dihedral angles to be sampled. The sampling protocols presented in Chapters 2 and 4 begin with Eq. 41.

The form of Eq. 44 suggests that a natural partition of coordinates (sidechain and backbone) exists, and that a sampling method that is cognizant of such partitions may provide improvements. Chapter 4 addresses this idea in considerable detail.

Chapter 2

Multiscale Monte Carlo Sampling of Protein Sidechains: Application to Binding Pocket Flexibility

Jerome Nilmeier

Graduate Group in Biophysics, University of California at San Francisco

Matt Jacobson

Department of Pharmaceutical Chemistry University of California at San Francisco

Abstract

We present a Monte Carlo sidechain sampling procedure and apply it to assessing the flexibility of protein binding pockets. We implemented a multiple 'time step' Monte Carlo algorithm to optimize sidechain sampling with a Surface Generalized Born implicit solvent model. In this approach, certain forces (those due to long-range electrostatics and the implicit solvent model) are updated infrequently, in "outer steps", while short-range forces (covalent, local nonbonded interactions) are updated at every "inner step". Two multi-step protocols were studied. The first protocol rigorously obeys detailed balance, and the second protocol introduces an approximation to the solvation term that increases the acceptance ratio. The first protocol gives a 10 fold improvement over a protocol that does not use multiple time steps, while the second protocol generates comparable ensembles, and gives a 15 fold improvement. A range of 50–200 inner steps per outer step was found to give optimal performance for both protocols. The resulting method is a practical means to assess side chain flexibility in ligand binding pockets, as we illustrate with proof-of-principle calculations on 6 proteins: DB3 Antibody, thermolysin, estrogen

receptor, PPAR-γ, PI3 kinase, and CDK2. The resulting sidechain ensembles of the *apo* binding sites correlate well with known induced fit conformational changes, and provide insights into binding pocket flexibility.

Introduction

Side chain sampling and optimization algorithms, mostly based on a rotamer approximation[22-26], have been used extensively in modeling proteins, including homology modeling[27,28] and predicting conformational changes due to ligand binding[29-31]. We have been interested in developing sampling methods for protein side chains (and, in other work, loops) that generate thermodynamic ensembles of conformations, in contrast to locating the global energy minimum[32,33]. Minimization methods implicitly neglect the effect of entropy on side chain conformations, and generally cannot distinguish whether sidechains will adopt a single well-defined conformation, or a distribution of conformations. For the many sidechains that are tightly packed in the core of a protein, minimization is an effective approach. For less tightly packed sidechains that display some degree of flexibility, a thermodynamic ensemble becomes a more appropriate description.

Side chain conformational heterogeneity is important to protein-ligand binding. The ability to accurately predict the flexibility/rigidity of binding site residues would be useful in structure-based drug design[31,34]. For example, a recent paper by Sherman *et al.* [29] describes a computational method to predict "induced fit" effects upon ligand binding which relies on some advanced knowledge of which side chains may adopt different conformations upon ligand binding, e.g., from multiple co-crystal structures. We demonstrate here that thermodynamic ensembles of side chain conformations in *apo* proteins correlate well with known induced fit conformational changes in various well studied drug targets.

In principle, molecular dynamics sampling methods [31,35] can be used to obtain thermodynamic ensembles for protein binding sites. The main disadvantage is that the timescales required to observe large changes in side chain conformations can be long relative to the ~1 fs timesteps employed in atomically detailed molecular dynamics simulations; transitions between side chain rotamers can take up to μs, which is a known difficulty in binding affinity calculations [35-37]. Monte Carlo sampling [38] can lead to more efficient generation of the complete thermodynamic ensemble, if the trial moves are constructed carefully.

For macromolecules, which contain complex, heterogeneous, and densely packed atomic configurations, construction of efficient trial moves can be a substantial challenge. A variety of both rigorous and nearly rigorous methods have been used [21,33,39-43] to address this challenge. One common idea among these involves decomposing the degrees of freedom into subspaces that are more manageable, both computationally and conceptually. The most natural decomposition for proteins is between backbone and sidechain degrees of freedom. Future work will incorporate backbone motions, but the current emphasis is on the sidechain degrees of freedom.

Another common decomposition is between solvent (water) and solute (protein) degrees of freedom. Here we use an implicit solvent model, which makes it possible to efficiently sample large side chain conformational changes. By contrast, in explicit solvent, large changes (e.g., across rotamers) are difficult to sample with good acceptance rates because of steric clashes between waters and the side chain, and the need for the solvent to relax around any new trial conformation. The same steric issues have motivated the use of implicit solvent in molecular dynamics studies as well[44-46]. For this work, the electrostatic solvation term is evaluated with the SGB model [20,47] and the nonpolar solvation energy with the nonpolar (NP) model [48]. The solvation model here was developed for use with the all atom OPLS-AA 2001 forcefield [49] and is implemented in the Protein Local Optimization Program [50,51]. While this model is chosen as a compromise between efficiency and accuracy, it remains the most computationally expensive portion of the energy evaluations. The current effort is to develop a general sampling scheme which allows optimal use of an implicit solvation model in the context of a Monte Carlo scheme. The present application is to sidechain sampling, but can be extended to backbone sampling strategies in a straightforward manner.

The major innovation here in terms of computational methods is the implementation of a multi-scale strategy, analogous to methods such as RESPA [52,53], used in molecular dynamics, to accelerate convergence toward the thermodynamic ensemble. The theory underlying this approach has been presented previously [54], and is only briefly reviewed here. The application of a multiscale Monte Carlo approach to sampling proteins in implicit solvent has been presented by Michel *et al*[55], with different implementation details and approximations introduced. Other algorithmic details crucial for speed, including the rapid elimination of conformations with steric clashes, are also described. The resultant method is a practical means to assess side chain flexibility in ligand binding pockets, as we illustrate with proof-of-principle calculations on 6 proteins.

Theory and Methods

Configuration Integral

The implicitly solvated [19] macromolecular ensembles of interest can be represented by the following configuration integral:

Equation 45

$$Q = \int d\mathbf{R} \exp\left(-\beta[A(\mathbf{R})]\right)$$

where is **R** is the set of all Cartesian coordinates of the macromolecule of interest, and

Equation 46

$$A(\mathbf{R}) = U(\mathbf{R}) + G(\mathbf{R})$$

where $A(\mathbf{R})$ is the sum of the forcefield energy, $U(\mathbf{R})$, and the implicit solvation energy, $G(\mathbf{R})$. The solvation energy is dependent on the Born radii, which are a function of the coordinate state of the macromolecule. In the SGB implementation we use, the Born radii $\alpha(\mathbf{R})$ are computed using surface integrals, and thus are dependent on the global coordinate state **R** of the protein. This calculation can take much longer (roughly 100 times longer in cases studied) than the pairwise energy terms. Some improvements have been gained by updating only local regions of the surface area as needed, and efforts are ongoing in this area to improve the efficiency and accuracy of this model [56,57].

In general, however, any attempt to optimize sampling would benefit most from evaluating the solvation energy less frequently. While this approach is motivated by computational efficiency, a physical argument can also be made. The Born radii generally vary slowly for relatively small, local conformational changes. The sampling strategies presented are intended to make the best use of these ideas while still generating meaningful ensembles.

Constraints on various degrees of freedom can be introduced to generate a configuration integral q_0 over a smaller subspace by identifying fixed (F) and sampled (S) degrees of freedom, such that $d\mathbf{R}=d\mathbf{R}^{<F>}d\mathbf{R}^{<S>}$, and imposing a rigid constraint on the fixed degrees of freedom, yielding-

Equation 47

$$q_0 = \int d\mathbf{R}^{<S>} \exp\left(-\beta A[(\mathbf{R}^{<S>} \mid \mathbf{R}_0{}^{<F>})]\right)$$

Following the formulation of Deem [21], the transformation from Cartesian to torsional coordinates can be made with a Jacobian of unity, if bond lengths and angles are preserved. For the current work, the backbone torsions will be constrained to an initial value of $\boldsymbol{\varphi}_0$, and the fixed sidechains to an initial value of $\boldsymbol{\chi}_0{}^{<F>}$. The resulting integral can be recast as

Equation 48

$$q_0 = \int d\boldsymbol{\chi}^{<S>} \exp\left(-\beta[A(\boldsymbol{\chi}^{<S>} \mid \boldsymbol{\varphi}_0, \boldsymbol{\chi}_0{}^{<F>})]\right)$$

where $\boldsymbol{\chi}^{<S>}$ is the set of sidechain torsional coordinates that are sampled. The integral of interest over the subspace can be recast by letting $\mathrm{d}\mathbf{r}=\mathrm{d}\mathbf{R}^{<S>}$, and $A(\mathbf{r}) = A(\boldsymbol{\chi}^{<S>} \mid \boldsymbol{\varphi}_0, \boldsymbol{\chi}_0{}^{<F>}) = A(\mathbf{R}^{<S>} \mid \mathbf{R}_0{}^{<F>})$, yielding the more compact expression:

Equation 49

$$q = \int d\mathbf{r} \exp\left(-\beta[A(\mathbf{r})]\right)$$

Generation of Trial Configurations

To generate a reversible trial move, a single sidechain *i* is chosen at random from the list of sampled sidechains, and the updated set of torsions is assigned according to:

Equation 50

$$\chi'_i = \chi_i + \xi$$

where χ'_i and χ_i are the trial and previous set of dihedral coordinates, respectively, for sidechain *i*, and ξ is a vector of uniform random variates of the same dimension, for which each value is drawn from the domain [-*d*/2, *d*/2]. To account for local fluctuations as well as larger fluctuations, the domain size *d* is assigned a value of either 360º or 18º with equal probability. The idea behind the heterogeneous move set is to alternate between large dihedral trial moves that cross local χ wells, and small trial moves, which sample the local χ basin. For the present work, selections from a rotamer library are not incorporated as a trial move, as slight nonuniformities in the distribution of the χ angles of the rotamer library have a quantitative effect on the distributions. As a practical matter, however, a mixture of rotamer and random moves could conceivably be implemented if quantitative energy distributions are not required.

For residues with rotatable polar hydrogen groups (Cys, Ser, Thr, Tyr), the torsional angle that places the hydrogen is also selected randomly when the rotamer state is assigned. Also, the torsions of the amine hydrogens of lysines are sampled. Torsions for methyl hydrogens are not currently sampled.

A hard sphere approximation is invoked, which vastly improves sampling efficiency, while preserving much of the essential physics of the system. This has been shown in liquid systems [58,59] as well as proteins. For the current work, pairs of atoms that are closer than 0.7 times the sum of the Lennard-Jones radii are considered to be sterically disallowed. That is, no energy is computed for sterically disallowed states, because the steric clash will result in high energies and small acceptance probabilities. Cell lists (linked lists) further accelerate the identification of steric clashes, by only checking for clashes between atoms known to be proximal. A series of dihedral perturbations is generated as described until a configuration that is sterically allowed is generated. The resulting configuration is treated as a trial move. For the systems studied, the average number of sterically disallowed moves ranges from 0.5 to 0.75 (see Table 2.2), which is roughly a 2 to 4 fold improvement in sampling efficiency, because the CPU time per steric clash evaluation is negligible relative to the energy evaluation.

Multiple Time Step Monte Carlo (MTS-MC)

A sampling procedure known as multiple time step Monte Carlo[54], which was originally developed for Ewald sum calculations[60], can be used to optimally sample against a potential that can be decomposed into additive components. These components are typically, but not necessarily, short and long range contributions to the energy. The algorithm relies on the assumption that the short range term varies rapidly with respect to the move set, while the long range term varies more slowly. A related formalism is presented using approximate potentials [61]. Many algorithms use similar ideas, including both molecular dynamics integrators [52,53] and minimization algorithms [62]. Some applications using algorithms that are similar in spirit involve evaluating Ewald sums less frequently in fluid simulations with periodic boundary conditions, sampling of polar fluids [5], and polarizable water sampling [63].

While the formalisms in these approaches vary, they can all be thought of as relying on some decomposition of the overall potential to be sampled. The natural choice of decomposition is into short and long range terms, which we denote by subscripts *S* and *L*, respectively

Equation 51

$$A(\mathbf{r}) = A_S(\mathbf{r}) + A_L(\mathbf{r})$$

The details of the nature of the decomposition of interactions into long and short range can vary from system to system. A more detailed description of the decomposition for the present case, with proof of detailed balance, is given in the Supplementary Material.

Using the above decomposition, detailed balance can be maintained using the following sampling protocol:

1) Starting with the configuration $\mathbf{r}_i$, generate a number N_I of 'inner loop' steps, where each step consists of a trial configuration $\mathbf{r}_k$ that is generated reversibly (such as the trial configurations described by Eq. 50, and accepted according to the following short range acceptance criterion:

Equation 52

$$\frac{acc_S(\mathbf{r}_{k'} \mid \mathbf{r}_k)}{acc_S(\mathbf{r}_k \mid \mathbf{r}_{k'})} = \exp\left(-\beta[A_S(\mathbf{r}_{k'}) - A_S(\mathbf{r}_k)]\right)$$

2) Take the final configuration from the inner loop to be the trial configuration $\mathbf{r}_j$ for the 'outer loop' and apply the long range acceptance criterion:

Equation 53

$$\frac{acc_L(\mathbf{r}_j \mid \mathbf{r}_i)}{acc_L(\mathbf{r}_i \mid \mathbf{r}_j)} = \exp\left(-\beta[A_L(\mathbf{r}_j) - A_L(\mathbf{r}_i)]\right)$$

It is important to note that any statistical quantities of interest can only be computed using the outer loop configurations. In all cases where the ratio of acceptance probabilities are given, the Metropolis acceptance criterion is used in practice.

Recasting MTS-MC to Account for Infrequent Born Radii Updates

For the present case, the most costly term to evaluate in the energy is the solvation term, which is due largely to the time intensive step of computing the Born radii, $\alpha(\mathbf{R})$, and we develop a strategy such that the Born radii are not updated in the inner steps. To motivate this method, it is helpful to express the potential in the following form:

Equation 54

$$A(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n) = U(\mathbf{r}_n) + G(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n)$$

where $\mathbf{r}_n$ is *n*th configuration of the subset of sampled coordinates, $\alpha(\mathbf{R}_m)$ **is** the set of Born radii which are evaluated based on the coordinates of the *m*th coordinate state $\mathbf{R}_m$ of the entire protein, $U(\mathbf{r}_n)$, and $G(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n)$ is the solvation energy evaluated at the given states. We can further express the energy deviation from the 'true' potential, where the Born radii are synchronous with the current coordinate state, in terms of an error potential $\varepsilon(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n)$:

Equation 55

$$\begin{aligned} \varepsilon(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n) &= A(\boldsymbol{\alpha}(\mathbf{R}_n), \mathbf{r}_n) - A(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n) \\ &= G(\boldsymbol{\alpha}(\mathbf{R}_n), \mathbf{r}_n) - G(\boldsymbol{\alpha}(\mathbf{R}_m), \mathbf{r}_n) \end{aligned}$$

Thus, the inner loop configurations are evaluated according to an approximate short range potential $A_S(\boldsymbol{\alpha}(\mathbf{R}_m),\mathbf{r}_n)$, where the Born radii are held at a previous, or 'latent' state. The relation to the true short range potential can similarly be written in terms of a short range error potential $\varepsilon_S(\boldsymbol{\alpha}(\mathbf{R}_m),\mathbf{r}_n)$:

Equation 56

$$A_S(\boldsymbol{\alpha}(\mathbf{R}_n),\mathbf{r}_n) = A_S(\boldsymbol{\alpha}(\mathbf{R}_m),\mathbf{r}_n) + \varepsilon_S(\boldsymbol{\alpha}(\mathbf{R}_m),\mathbf{r}_n)$$

where the coordinate state is $\mathbf{r}_n$, and the latent Born radii, $\boldsymbol{\alpha}(\mathbf{R}_m)$ are calculated from a previous step. Likewise, the true long range potential can be described in terms of long range error potential:

Equation 57

$$A_L(\boldsymbol{\alpha}(\mathbf{R}_n),\mathbf{r}_n) = A_L(\boldsymbol{\alpha}(\mathbf{R}_m),\mathbf{r}_n) + \varepsilon_L(\boldsymbol{\alpha}(\mathbf{R}_m),\mathbf{r}_n)$$

For simplicity, these energies can be expressed in terms of the state indices only:

Equation 58

$$\begin{gathered}
A_S(n,n) = A_S(m,n) + \varepsilon_S(m,n) \\
A_L(n,n) = A_L(m,n) + \varepsilon_L(m,n) \\
\varepsilon(m,n) = \varepsilon_S(m,n) + \varepsilon_L(m,n)
\end{gathered}$$

Where n is the index of the current coordinate state, and m is the index of the Born radii held at a previous state. We can simply recast the decomposition as:

Equation 59

$$\begin{gathered}
A(n,n) = A_S(n,n) + A_L(n,n) \\
= A_S(m,n) + \varepsilon(m,n) + A_L(m,n) \\
= A(m,n) + \varepsilon(m,n)
\end{gathered}$$

where the index of the coordinate state is first argument in each of the functions, and the index of the Born radii state is the second argument. While the error potential described in Eq. 14 contains both long and short range terms, the idea of the sampling protocols is to treat the all of error potential terms as long range terms. Using this new decomposition, we can define two different sampling protocols:

1) In both protocols, start with the configuration $\mathbf{R}_i$, generate a number N_I of 'inner loop' steps, where each trial configuration $\mathbf{r}_k$ is generated using Eq. 51. The Born radii are held at a latent state i, such that the short range acceptance criterion is:

Equation 60

$$\frac{acc_S(k'|k)}{acc_S(k|k')} = \exp\left[-\beta\left(A_S(i,k') - A_S(i,k)\right)\right]$$

2) Take the final configuration from the inner loop to be the trial configuration $\mathbf{r}_j$ for the 'outer loop' and apply either of 2 acceptance criteria:

A. With error correction:

Equation 61

$$\frac{acc_L(j|i)}{acc_L(i|j)} = \exp\left[-\beta\left(A_L(i,j) + \varepsilon(i,j) - A_L(i,i)\right)\right]$$

B. Without error correction:

Equation 62

$$\frac{acc_L(j|i)}{acc_L(i|j)} = \exp\left[-\beta\left(A_L(i,j) - A_L(i,i)\right)\right]$$

Protocol ***A*** rigorously obeys detailed balance, while protocol ***B*** is an approximation introduced to improve computational efficiency. It should be noted that the Born radii are completely updated in every outer loop calculation, regardless of protocol. The ideal error potential term would be narrowly distributed about a mean of zero, so that the distribution generated by neglecting the term would be nearly equivalent to the true distribution. The effect of the modification will be discussed in detail in the results section.

As a control, a "standard" Monte Carlo trajectory, or protocol ***S***, was also studied. For the standard Monte Carlo protocol, the same trial move set was used, including steric screening, but with the Born radii updated at every step, with no decomposition of potentials. For every step, the acceptance criterion is simply:

Equation 63

$$\frac{acc_S(j\,|\,i)}{acc_S(i\,|\,j)} = \exp\left[-\beta\left(A(j,j) - A(i,i)\right)\right]$$

Estimation of the Time to Convergence and Improvement Ratio

To estimate the optimal number of inner steps, we express the total processor time *T* to compute a trajectory as:

Equation 64

$$T = N_{O,T} < dt\,/\,dN_O >$$

where $< dt\,/\,dN_O >$ is the expectation value of the time required to generate an outer step. This is not a fixed value, since the innermost sampling loop samples an arbitrary number of configurations until a sterically allowed configuration is obtained. $N_{O,T}$ is the total number of outer steps, which includes the both the nonequilibrated steps, n_O, and equilibrated steps, N_O. This can also be expressed as:

Equation 65

$$T = N_{O,T}(t_L + N_I t_S)$$

where t_S is the average time required to generate a single (sterically allowed) trial coordinate and evaluate the short range potential. The rate t_L is the time required to evaluate the long range potential, which includes the long range energies and the time required to update the Born radii. This quantity does not need to be averaged, since there is no dependence on the number of steric clashes. N_I is the number of inner steps that are set for the simulation. Since statistics can only be gathered on the equilibrated outer steps, we can express N_O in terms of the standard error:

Equation 66

$$N_O = \frac{\sigma^2}{\varepsilon^2} g(N_I)$$

where σ is the variance of the energy over the entire equilibrated portion of the trajectory, ε is the desired error in the estimate of the energy, and $g(N_I)$ is the correlation interval, or distance between uncorrelated snapshots. This quantity is measured from the simulation, and will vary with the number of inner steps for a given system with all other conditions held constant. It is closely related to other measures of quality of Monte Carlo trajectories, such as acceptance ratio, and a low correlation interval often corresponds to a high acceptance ratio.

Since the number of steps required to equilibrate depends strongly on the initial condition, we shall overestimate this quantity by assuming that n_O=N_O. This varies in practice from a few correlation intervals to less than half of the number of outer steps. As long as the equilibration time is proportional to the number of equilibrated steps, it will cancel out in the improvement ratio calculation. Using this assumption, the estimated CPU time required for a converged trajectory is:

Equation 67

$$T = 2\frac{\sigma^2}{\varepsilon^2} g(N_I)(t_L + N_I t_S)$$

where the number of inner steps can be adjusted to locate the optimal computing time. As a measure of sampling efficiency, the following quantity can be expressed:

Equation 68

$$I = T_S / T$$

where I is the improvement, and T_S is the time required for a converged trajectory in a standard Monte Carlo protocol.

Convergence Determination and Error Estimation

Determination of the number of steps required for equilibration and the correlation interval was performed iteratively. Initially, the number of steps required for equilibration was estimated to be 3000 for the standard trajectory, 1000 for N_I =1,50,100,200, 300, and 400 for the remaining inner step settings. To estimate the correlation time, an autocorrelation function of the energy was computed, and the correlation interval g was identified as the first place the autocorrelation function crosses zero. This initial estimate is expected to overestimate the true correlation time since the trajectory may include nonequilibrated regions, which contain slow fluctuations towards the equilibrium state that would not be present in the stationary distribution. Using this initial estimate, a blocksize was assigned to have a value of g. A block standard deviation σ_B is computed at each point (using the points preceding the point of interest), and the trajectory was deemed to be converged if the block standard deviation was less than a nominal value σ_B=15k_BT.

With this new estimate of the equilibrated region of the trajectory, another estimate of the correlation time was applied. To improve the estimate, the autocorrelation function was fit to a simple exponential $\exp(-\tau / \tau_D)$ where τ_D is the decay constant, or correlation time. For this procedure, a least squares fit was performed where the sum of the squares of the errors between the function and the data points are weighted according to the inverse of error at that point. The error in the autocorrelation function is given by[5]:

Equation 69

$$\varepsilon\left[C(\tau)\right] = \sqrt{\frac{g}{N_O - \tau}}$$

where g=1+2τ_D is the correlation interval, or the number of steps between uncorrelated snapshots. Once a correlation time is obtained, the Reverse Cumulative Averaging method was used to obtain a better estimate of the location of the equilibrated region [64], with the blocksize set to g. A confidence level of 85% was used to reject the hypothesis that the block averaged samples came from a normal distribution, according to the Shapiro-Wilk Test [65,66]. The location of the equilibrated portion of the trajectory depends heavily on the value of the blocksize, and vice versa, so 30 iterations of the blocksize and RCA convergence calculation were run. See Figure 2.1 for the convergence times, correlation intervals, and total simulation lengths for each simulation.

N_I	$N_{O,T}$	<dt/$dN_{O,T}$ > (s)	<E>-<E>$_{STD}$ (RT)	σ	ε	N_O (all)
S	250000	6.02	0.00	5.65	0.37	2366928
A-1	95000	6.29	-0.19	5.88	0.76	421025
A-50	40000	12.02	-0.01	5.83	0.21	195235
A-100	25000	16.37	0.13	5.83	0.19	122849
A-200	15000	26.08	0.17	5.86	0.22	74035
A-300	10000	34.37	0.25	5.82	0.24	48860
A-400	6000	43.44	0.12	5.83	0.31	29354
A-500	5000	51.48	0.13	5.91	0.37	24195
B-1	95000	6.20	1.77	5.96	0.28	393587
B-50	40000	11.11	1.58	6.07	0.07	198311
B-100	25000	16.01	1.79	6.12	0.08	123416
B-200	15000	24.72	1.89	6.12	0.08	73904
B-300	10000	32.86	1.71	6.19	0.10	48913
B-400	6000	41.40	1.69	6.09	0.11	29440
B-500	5000	50.47	1.83	6.18	0.12	24568

Table 2.1 – Simulation Data for Model System. Data shown summarizes the results for 10 simulations of each Protocol and Inner step setting. For leftmost column, N_I is the number of inner steps. S indicates a standard protocol (no inner steps). For the remaining columns, protocol and number of inner steps are given. (A-50 represents protocol A using 50 inner steps). $N_{O,T}$ is the total number of steps simulated, including nonequilibrated portions of the trajectory. <dt/$dN_{O,T}$ > is the average time to generate an outer step, as described in the text. <E>-<E>$_{STD}$(RT) is the average equilibrium energy minus the standard measurement, σ and ε are the standard deviation and standard error of the equilibrated energies. Rightmost column is the total number of equilibrated steps (across all simulations at the designated setting) used for the calculation.

Preparation of Unbound receptors

The proteins studied are listed in Table 2.2. A few of the proteins had missing side chains or loops outside of the binding sites (>15Å) being studied. These were reconstructed in arbitrary configurations free of steric clashes using standard routines in the Protein Local Optimization program. The side chains to be sampled in the Monte Carlo were defined as those within 8 Å of any atom of the ligand in the *holo* structure. All calculations were performed in the absence of the ligand.

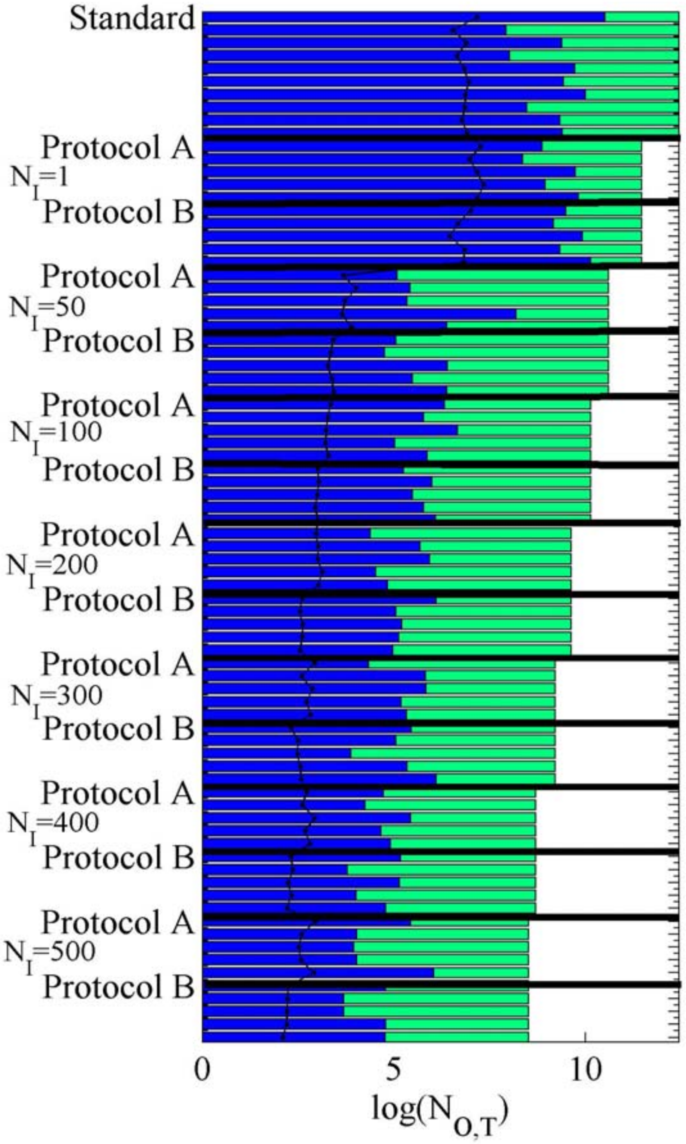


Figure 2.1. Summary statistics for validation dataset. Bars represent log of simulation lengths, and black dots connected with lines represent the correlation interval for that simulation. All simulations are run at 600K. The blue portion of each bar is the unequilibrated portion, and the green portion is equilibrated. Different values are given for different runs, which are trajectories using the same settings, including initial condition, but assigned different random seeds. The natural log of the number of Total steps, $N_{O,T}$, appears on the x-axis.

Composite Energy Histograms

In order to represent multiple simulations of the same sampling protocol as a single histogram, a superposition of individual energy histograms was computed. This is done to obtain better statistics so that detailed balance may be demonstrated for protocol.

For each trajectory histogram, an error $\varepsilon_B = \sqrt{gn_B}$ was assigned at each bin point, where n_B is the number of entries in each bin. To generate the composite histograms for protocols ***A*** and ***B***, each of the trajectory histograms for each protocol were superimposed with a weight proportional to the number of uncorrelated entries in each bin of each trajectory. The errors are computed a superposition of square of the errors of each trajectory, with the same weights used to compute the composite histograms. It should be noted that the sampling protocols produce the same distribution of energies, independent of number of inner steps chosen. The data from all ranges of inner steps can therefore be combined to form a single histogram. Since the error is computed using the autocorrelation times, the fact that the distributions fall within error suggest also that the correlation times are correctly estimated.

Timings

Since simulations were run on a variety of machines, smaller trajectories were collected to estimate the average time per outer step (see Table 2.1). Timings of the simulations were measured on a Linux machine, using a single CPU from a dual AMD Opteron CPU running at 2.2 GHz.

Results and Discussion

Comparison of Protocols Using Antibody DB3

To optimize the number of inner steps and other parameters of the algorithm, the binding pocket of *apo* antibody DB3 (1dba) [67,68] was selected as a model system. A total of 3 sampling protocols were explored, as defined in Methods. To compare the effect of neglecting the short range error in the Born updates, identical simulations were run using protocols ***A*** (rigorous) and ***B*** (approximate). A single set of 10 trajectories using protocol ***S*** was also generated. The number of inner steps (N_I) was set to 1, 50, 100, 200, 300, 400, and 500. For each inner step setting, 5 trajectories were collected, starting from the same (nonequilibrium) initial condition with different random seeds. Since the backbone is held fixed, room temperature simulations tend to exhibit frustrated dynamics. To obtain better statistics, especially for protocol ***S***, all simulations were run at 600 K. The goals of these simulations are twofold: 1) to generate sufficient statistics to demonstrate detailed balance, and 2) to study the effect of adjusting the number of inner steps and protocol. A total of 80 separate trajectories were collected for the analysis. Figure 2.1 summarizes the pertinent information on these trajectories.

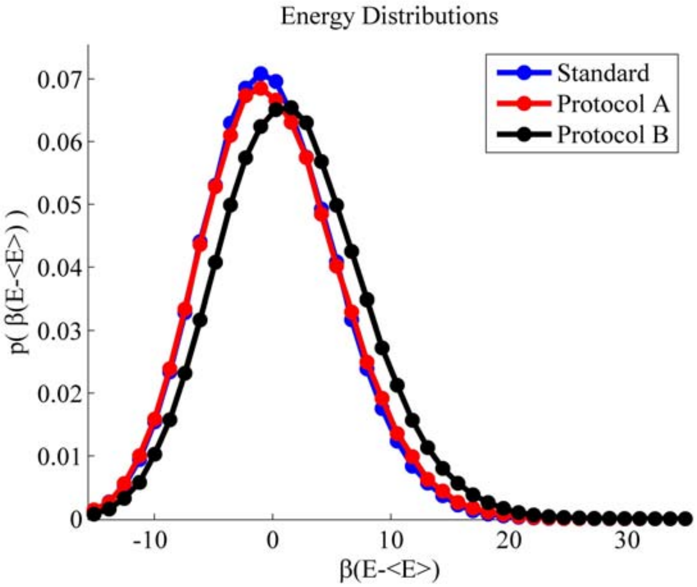


Figure 2.2. Protocol A distributions superimpose with Standard energy histograms, and Protocol B generates a similar approximate distribution. All simulations were run at 600K, under the conditions summarized in Figure 2.1. Dimensionless energy is plotted on the x axis, with the mean of the energies of the standard simulation <E> subtracted from the energy (see Table 2.1). On the y axis is the probability of observing that energy.

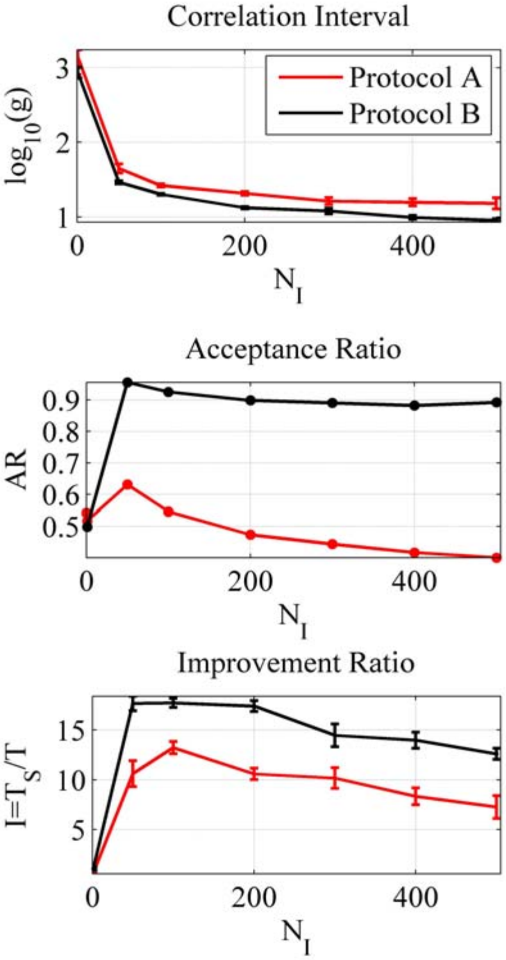


Figure 2.3. Approximate protocol provides slightly better performance, and optimal performance of both protocols is in the range of N_I=50-200. a) log of correlation interval, b) acceptance ratio, c) improvement ratio, as given by Eq. 68.

The average energies and standard errors of each simulation are in Table 2.2, and Figure 2.2 shows histograms of equilibrated energies for each sampling protocol. The energy distributions of protocols ***A*** and ***S*** (standard) appear to be equivalent. While error bars are not shown for clarity, the histograms superimpose to well within the estimated error. The energy distribution of protocol ***B*** is offset by roughly 1.75RT, and is clearly from a different distribution than protocol ***A***. The standard deviation of protocol ***B*** is larger by roughly 0.3RT. The broader distribution and higher mean value is due to the more permissive approximation, which increases the number of states that are accepted. The correlation interval is shown in Figure 2.3. A sharp decrease is observed from N_I = 50–200**,** which steadily decreases over the remaining inner step settings. The acceptance ratio shows an initially sharp increase, since a smaller number of inner steps helps to generate better trial moves for the outer loop. As the number of inner steps increase however, the inner loop becomes less efficient at generating trial configurations. This effect is more prominent in protocol ***A***, which is the rigorous approach. Figure 2.3c shows the relative improvement over protocol ***S*** (no inner steps). Optimal values are in the range N_I = 50–200. For both protocols ***A*** and ***B***, a broad optimal range is observed, which suggests that this optimal range should hold for a wide variety of proteins.

Binding Pocket Studies

As a first application, we investigate the flexibility of side chains in protein binding pockets. As a test set, we consider several proteins from Sherman *et al*[29], as well as PI3K[69]. The assumption of this work is that side chains that show more flexibility in our ensembles will be capable of undergoing rearrangements upon binding ligands. Table 2.2 lists the binding pockets studied. For all trajectory data which is displayed, individual sidechains conformations were filtered such that no two conformations are less than an RMSD of 0.05 Å from one another.

Protocols ***A*** and ***B*** were used to generate side chain ensembles, at a variety of temperatures. Temperatures >300K were explored for three primary reasons. First, our goal is to predict conformational changes that could occur upon binding a ligand. In the limit of pure "conformational selection", the bound conformation of the protein would be populated significantly, or at least measurably, at ambient temperature. However, there can also be some additional conformational rearrangement of the 1protein to accommodate the ligand ("induced fit"), derived from the free energy of ligand binding. Here, we have essentially postulated that ligand binding can "induce" conformational changes that may not be observable with a room temperature thermal ensemble. It has been observed that sidechain rearrangements within binding pockets can be cost up to 4 kcal/mol of free energy [36,37].

Another reason for considering higher temperature distributions of 600 K is related to limitations of the energy function. In particular, it has been widely reported that Generalized Born solvent models can over-stabilize hydrogen bonds and salt bridge interactions[57,70]. This known limitation of the implicit solvent model will tend to result in reduced flexibility of charged residues at ambient temperatures.

Finally, the use of a rigid backbone will also reduce side chain flexibility. The test cases were chosen in part because ligand binding does not induce large changes in backbone conformation; clearly, further algorithmic development, which will be reported in due course, is needed to deal with backbone fluctuations. When there is reason to believe that backbone changes are likely to be small, simply using a higher temperature may help to reduce artifacts due to the rigid backbone.

Ultimately, from the standpoint of identifying "flexible" side chains in a binding site, we view the choice of temperature as a user-definable parameter; in practice, performing simulations with multiple values of the temperature may be advisable. Note that, since the backbone is held fixed, the protein will not denature during the simulation, which provides considerable freedom in the choice of temperature and simulation protocol.

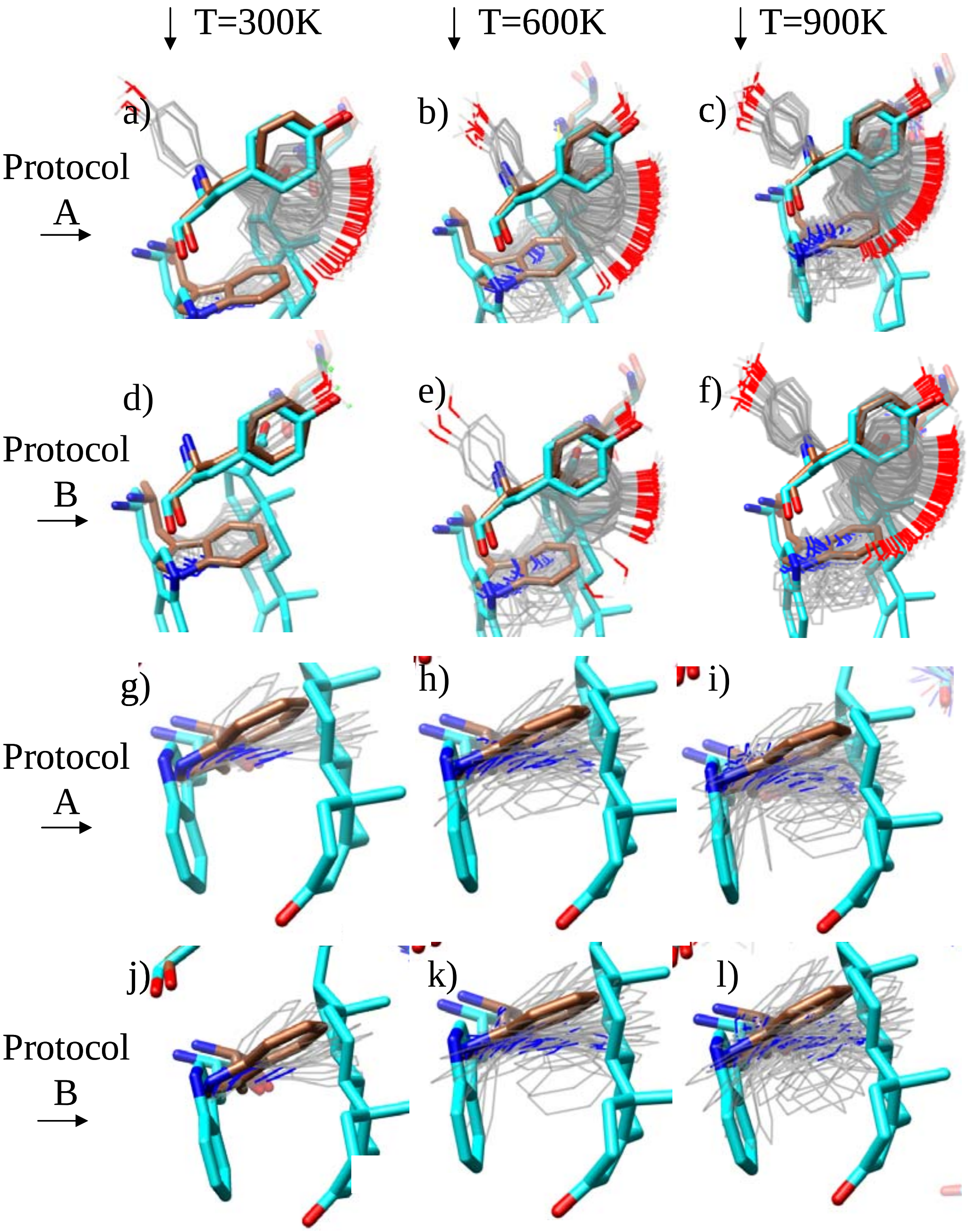


Figure 2.4. Distribution of sidechain configurations for Tyr97 and Trp100 of 1dba. Brown configurations are from the native structure, cyan configurations are from the *holo* structure. Grey sidechains are distinct configurations from a sidechain trajectory at the given conditions. a) Tyr97 at 300K, protocol A; b) Tyr97 at 600K, protocol A; c) Tyr97 at 900K, protocol A; d) Tyr97 at 300K, protocol B; e) Tyr97 at 600K, protocol B; f) Tyr97 at 900K, protocol B; g) Trp100K at 300K, protocol A; h) Trp100 at 600K,

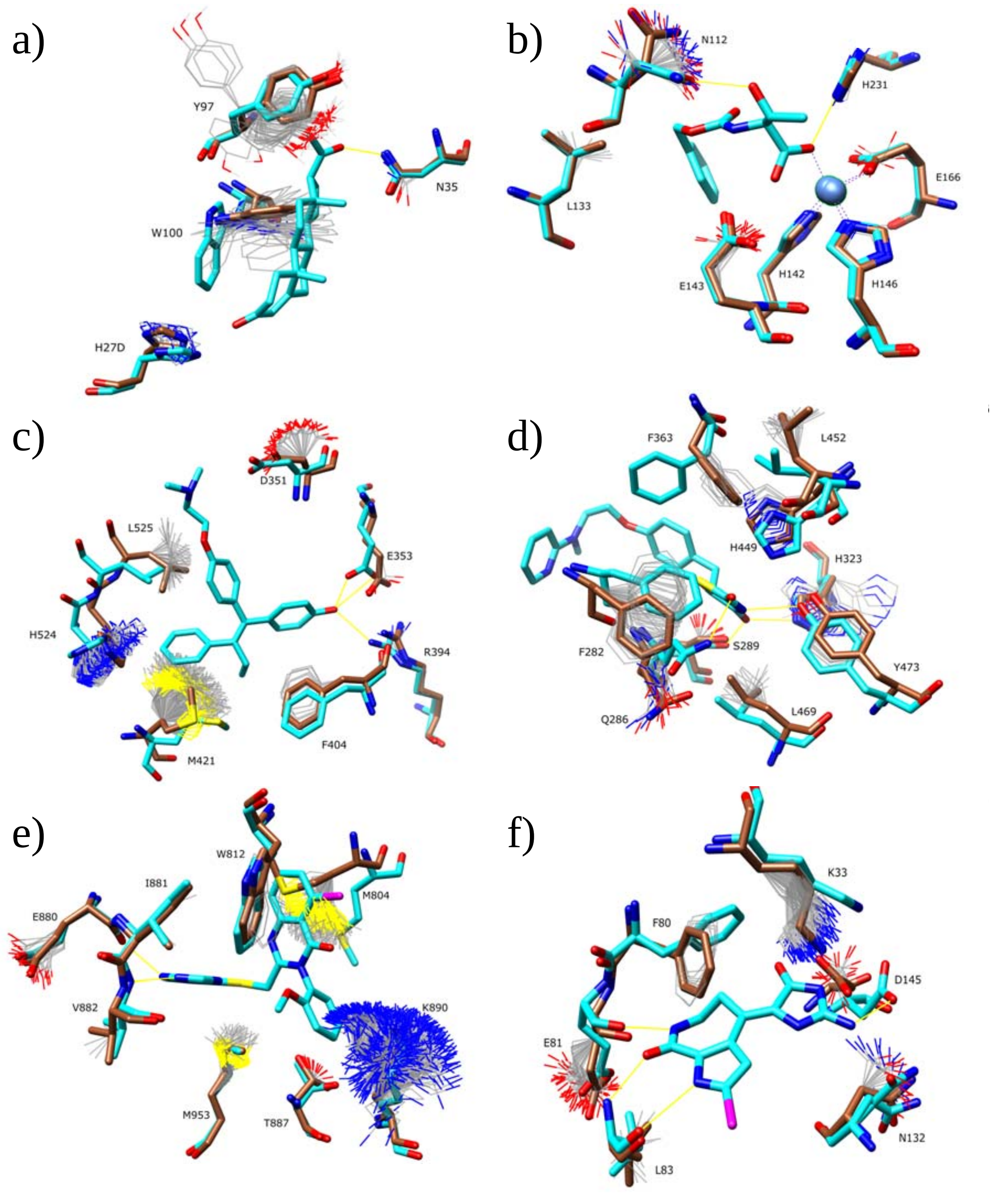


Figure 2.5. Binding pocket ensembles. Simulations are carried out in the absence of ligand at 600 K, with protocol B (no error correction). Ligand and bound (*holo*) structures are shown in cyan. Unbound native sidechains in starting configurations are shown in brown. The computed ensemble is shown as thin lines. The ligand from the holo structure is shown for reference. a) DB3 antibody and progesterone, b)where it is similar to the *apo* structure, although significant fluctuation is observed. Intermediate conformations are not observed suggesting a high energy barrier for the rotation.

Antibody DB3 [67,68]

For the DB3 antibody (Figures 2.4 and 2.5a), the primary conformational change between the two structures is the large movement of the Trp100 side chain to accommodate 4-hydroxytamoxifen. We studied this system with both protocols ***A*** and ***B*** at T=300, 600, and 900 K, with N_I=200 (the upper end of the optimal range). It is encouraging to observe that the large conformational change in Trp100 is observed in the Monte Carlo simulations, performed without a ligand present, at 600 K using protocol B and at 900 K using protocol A. Two conformational states of Trp100 are observed: a low-population state where the side chain is in a similar conformation as the *holo* structure, and a high-population state where the sidechain occludes the binding region.

Label	Protein	R_B	R_A	L_A	# residues	$<N_C>$
A	DB3 Antibody	1dba	1dbb	Progesterone	30	0.54
B	Thermolysin	1kr6	1kjo	Z-D Glutamic Acid	41	0.74
C	Estrogen Receptor	1err	3ert	Raloxifene	73	0.65
D	PPAR-γ	1fm9	2prg	GI262570	65	0.75
E	PI3 Kinase	2chx	2chw	PIK-039	45	0.65
F	CDK2	1buh	1dm2	hymenialdisine	46	0.73

Table 2.2 – Binding Pockets Studied. R_B is the receptor used in the simulation (without ligand), and R_A is a reference receptor with L_A bound to it. $<N_C>$ is defined as the total number of steric clashes divided by the number of sterically allowed steps.

The residues His27D and Asn35 show less flexibility in the simulations, and also little conformational change between the apo and holo structures (Figure 2.5a). Tyr97, by contrast, appears to fluctuate in multiple basins. This is because it is mostly solvent exposed, and there is very little steric hindrance. The side chain adopts similar conformations in the apo and holo structures. This does not necessarily imply a failure of the computational prediction, however. It is possible that this side chain could adopt different conformations in complex with other ligands.

The magnitudes of fluctuations observed using protocols ***A*** and ***B*** for Trp100 and Tyr97 are similar (Figure 2.4). Since protocol B is slightly more efficient and appears to provides similar configurational diversity, it was used for the data presented for all the remaining binding pockets in Figure 2.5. In addition, we have chosen to use T=600 K for the remainder of the test cases, because it provides a balance between sampling alternative conformations that may be important in ligand binding, but not so much diversity as to be uninformative. We reiterate that we view temperature as a user-adjustable parameter, and using multiple temperatures, as with this test case, may be advisable.

Thermolysin [71]

The residues His142, His146, and Glu66, which coordinate the Zn ion are correctly predicted to be rigid (Figure 2.5b). For this simulation, the zinc ion was included. The hydrogen bonding network of His231 is correctly preserved. Asn112 is predicted to be very flexible, and in fact rotates significantly upon ligand binding.

Estrogen Receptor [72,73]

Residues Leu525, Met421, and His524 all show significant flexibility in the simulations, and also undergo significant rearrangements upon binding 4-hydroxytamoxifin (Figure 2.5c). Glu353 and Arg394 display less flexibility due to the strong salt bridge. These show small conformational rearrangements upon binding the ligand due to formation of hydrogen bonds to it. Backbone rearrangements observed upon ligand binding, such as those seen in His524 and Leu525, are of course not captured by the side chain MC simulations. As a rough guide, however, the ensemble correlates well with observed rearrangements.

PPAR-γ [74]

The hydrophobic residues Phe282, Leu452, and Leu469 display flexibilities that correspond to structural rearrangements upon ligand binding (Figure 2.5d). Phe363 fails to sample the bound configuration, and is the first of only two false negative cases from the entire dataset (see CDK2). It is likely that this is due to the fact that the rigid backbone occupies a region which occludes the possibility of sampling an alternative state. His449 displays a narrow range of flexibility which corresponds to the displacement in the target structure. Tyr473 samples alternative solvent exposed configurations, similar to Tyr97 in the DB3 antibody. Gln286 displays flexibility, and appears to sample some conformations similar to the holo conformation, to the extent that the slightly different backbone configurations permit.

PI3 Kinase [69,75]

All residues which do not undergo significant rearrangement upon ligand binding are predicted to be rigid in the simulations (Figure 2.5e). Glu880 and Lys890 display conformational diversity in the

simulations which encompasses the observed apo and holo conformations. Met804 displays significant flexibility in the sidechain ensemble which encompasses the apo and holo conformations. The movement of this side chain is critical for opening a hydrophobic pocket that is critical for ligand binding and specificity.

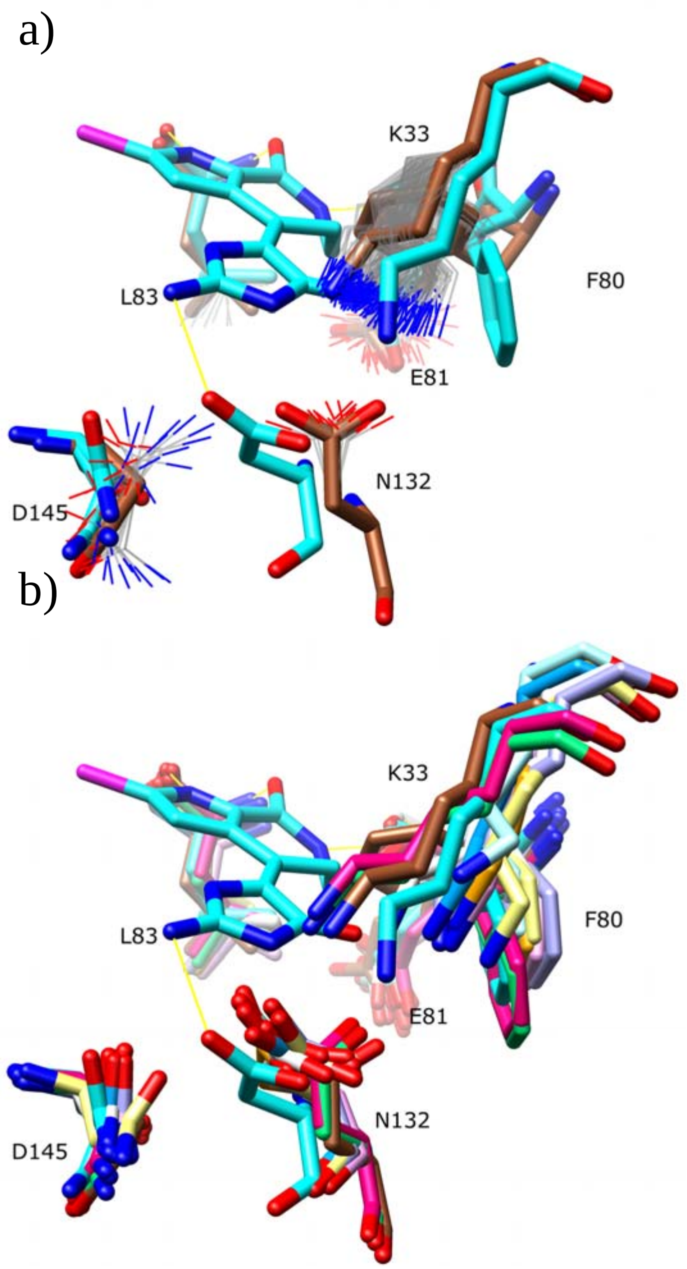


Figure 2.6. CDK2 salt bridge interaction. a) Binding pocket ensemble and representation is identical to Figure 2.5f, but from a different perspective. b) sidechains from CDK2 structures 1h24, 1h25, 1h26, 1h27, 1h28, 1hcl, 1pw2, 1w98, and 2jgz.

CDK2 [76-78]

Residues Glu81, Leu83, and Asn132 each appear to display conformational diversity commensurate with the observed changes between the apo and holo structures

(Figure 2.5f), while Phe80 is the second false negative of the dataset. Lys33 displays flexibility, although it does not quite sample the bound configuration. Instead, in the absence of ligands, it forms a salt bridge with Asp145, which is disrupted by the hymenialdiside interaction in the bound form.

Figure 2.6a shows a closeup of the salt bridge which is transiently disrupted in the 600 K simulation. Figure 2.6b shows a superposition of multiple structures of CDK2 which display a similar structural diversity.

Conclusions and Future Directions

A novel application of the MTSMC algorithm has been applied to sampling sidechain degrees of freedom in implicit solvent. Relative to a “simple” Monte Carlo algorithm without the use of inner steps, the multi-scale approach increases the convergence by a factor of 10–15. Rapid steric screening provides an additional factor of 2–4 speed up, and other algorithmic details (rapid updates of energies when only a portion of the protein is moving) also contribute to efficiency. Applications to small molecule ligand binding sites in proteins demonstrate that the method can be used to efficiently sample large changes in side chain conformations, and identifies side chains that may undergo conformational changes upon ligand binding.

Additional degrees of freedom can be incorporated into this approach in a straightforward manner. For example, local changes in backbone conformation can be included using analytical loop closure[79,80] methods with an appropriate Jacobian[81]. Such a method, which is under development, could be an efficient means of sampling conformational changes such as those that have been observed in the kinase DFG motif, or in loop latching as in TIM barrels[82], in a way that obeys detailed balance and thus can capture entropy differences between states.

Acknowledgments

We thank Kevan Shokat (UCSF) for pointing out the PI3K system. All molecular figures were generated with UCSF Chimera[83]. This work was supported by NSF grant MCB-0346399, NIH grant GM081710, and the Sandler Program in Basic Science (UCSF). MPJ is a consultant to Schrödinger, Inc. Special thanks to Katarzyna Bernacki, Dave Mobley, Arjun Narayanan and Chris McClendon for helpful discussions.

Appendix A : Proof of Detailed Balance with a Short Range Cutoff

A more detailed accounting of the short and long range decompositions is presented. These details omitted from the body of the text for clarity.

The use of a short and long range cutoff is a common way of improving calculation efficiencies. The advantage gained is in the infrequent updating of the long distance interactions. To explicitly track the updating of the short and long range cutoffs, Eqs. 58 and 59 can be re-expressed as:

Equation 70

$$A(m,n) = S(l)A(m,n) + (1 - S(l))A(m,n)$$

where *S(l)* is a ‘switching function’ of the coordinate state *l*, which divides the space over which the potential *A*(*m,n*), as expressed in equation 15, is the potential at Born state *m* and coordinate state *n*. When the Born radii are evaluated based on the current coordinate state, the short and long range potentials can be expressed in terms of the current coordinate (and Born radii) state *n*, and latent cutoff state *l*:

Equation 71

$$A_S(l,m,n) = S(l)A(m,n)$$
$$A_L(l,m,n) = (1 - S(l))A(m,n)$$

Since *S*(*l*) is a function of the complete set of coordinates, a full update of the distances must be computed. The idea behind the use of the cutoff is to limit the number of times the full distance matrix is computed, as well as the full potential.

To this end, an efficient Monte Carlo protocol will update the switching function infrequently, while maintaining detailed balance or very nearly doing so. For the updating scheme that is used for the present work, detailed balance is rigorously maintained with regard to the short and long range evaluations. The simplest form that the switching function can take is a simple distance cutoff, but more complicated forms, such as cell neighbor lists and other types of additive decompositions can be used. For this work, atoms are treated as short range if any single atom within a sidechain is within a cutoff distance of another sidechain. Default Settings that were developed for an optimal minimization strategy were used [62]. The cutoffs vary according to type of interaction. Each sidechain is identified as either charged or nonpolar. All atoms in the given sidechain are labeled as such. For nonpolar atoms interacting with nonpolar atoms, the cutoff is 15Å. For charged-nonpolar interactions, the cutoff is 20Å, and for charged-charged interactions the cutoff is 30 Å. The updating scheme used for the current work is to update the switching function at the beginning of the each ‘outer’ iteration of the sampling loop.

While the proof of detailed balance for the switching function updating scheme is independent of the Born radii updating scheme, the full bookkeeping of all latent states is presented here for completeness. Re-expressing the short and long range potentials in Eq. 56 with the short range state made explicit gives:

Equation 72

$$\begin{gathered} A_S(l,m,n) = A_S(l,n,n) - \varepsilon_S(l,m,n) \\ A_L(l,m,n) = A_L(l,n,n) + \varepsilon_L(l,m,n) \\ \varepsilon(m,n) = \varepsilon_S(l,m,n) + \varepsilon_L(l,m,n) \end{gathered}$$

The resulting (unnormalized) probability distributions are:

Equation 73

$$\begin{gathered} p_S(l,m,n) = e^{-\beta A_S(l,m,n)} \\ p_L(l,m,n) = e^{-\beta A_L(l,m,n)} \\ p_\varepsilon(m,n) = e^{-\beta \varepsilon(m,n)} \end{gathered}$$

Expressing the probability of a single state in terms of the decomposed states gives:

Equation 74

$$\begin{gathered} p(n) = e^{-\beta A(n,n)} / q \\ p(n) = p_S(l,n,n) p_L(l,n,n) = p_S(l,m,n) p_\varepsilon(m,n) p_L(l,m,n) \end{gathered}$$

Following the derivations presented in [54,61], the required detailed balance condition is:

Equation 75

$$\begin{gathered} p(i)T(j \mid i) = p(j)T(i \mid j) \\ p_S(i,i,i) p_L(i,i,i) T(j \mid i) = p_S(j,j,j) p_L(j,j,j) T(i \mid j) \end{gathered}$$

where *T*(*j*|*i*) is the probability of transitioning from coordinate state *i* to *j*. Expanding this expression gives:

Equation 76

$$\begin{aligned} & p_S(i,i,i)p_L(i,i,i)\alpha(j\,|\,i)acc_L(j\,|\,i) \\ & = p_S(j,j,j)p_L(j,j,j)\alpha(i\,|\,j)acc_L(i\,|\,j) \end{aligned}$$

where $\alpha(j|i)$ and $acc_L(j|i)$ are the selection and acceptance probabilities 'outer' state *j* from state *i*. Following the MTSMC derivation[54], the probability of selecting state *j* from state *i* is probability given by the following:

Equation 77

$$\alpha(j\,|\,i) = T_S^{(N_I)}(j\,|\,i)$$

where the above transition probability is the product of the individual transition probabilities of the inner loop:

Equation 78

$$T_S^{(N_I)}(j\,|\,i) = T_S(1\,|\,i)\left[\prod_{k=1}^{N_I-2} T_S(k+1\,|\,k)\right]T_S(j\,|\,N_I-1)$$

In the short range, or inner loop of sampling, neither the switc hing function nor the Born radii are updated, so that each step obeys the following detailed balance relation:

Equation 79

$$p_S(i,i,k)T_S(k'\,|\,k) = p_S(i,i,k')T_S(k\,|\,k')$$

The transition between outer states *j* and *i* obey the following detailed balance relation:

Equation 80

$$p_S(i,i,i)T_S^{(N_I)}(j\,|\,i) = p_S(i,i,j)T_S^{(N_I)}(i\,|\,j)$$

Combining eqs 32-35, and solving for the ratio of acceptance probabilities gives:

Equation 81

$$\left.\frac{acc_L(j\,|\,i)}{acc_L(i\,|\,j)}\right|_{TRUE} = \frac{p_L(j,j,j)p_S(j,j,j)}{p_L(i,i,i)p_S(i,i,j)}$$

Protocols ***A*** and ***B*** follow the same updating scheme for the switching functions. The acceptance probability for protocol ***A*** is expressed in Eq. 60 as:

Equation 82

$$\left.\frac{acc_L(j\,|\,i)}{acc_L(i\,|\,j)}\right|_{A} = \frac{p_L(i,i,j)p_\varepsilon(i,j)}{p_L(i,i,i)}$$

The ratio of Eqs. 81 and 82 is unity:

Equation 83

$$\begin{aligned}&\left(\left.\frac{acc(j\,|\,i)}{acc(j\,|\,i)}\right|_{TRUE}\right)\Bigg/\left(\left.\frac{acc(j\,|\,i)}{acc(j\,|\,i)}\right|_{A}\right)\\ &= \frac{p_L(j,j,j)p_S(j,j,j)}{p_L(i,i,i)p_S(i,i,j)}\cdot\frac{p_L(i,i,i)}{p_L(i,i,j)}\\ &= \frac{p_L(j,j,j)p_S(j,j,j)}{p_S(i,i,j)p_\varepsilon(i,j)p_L(i,i,j)} = \frac{p(j)}{p(j)} = 1\end{aligned}$$

and therefore the sampling scheme described by Eqs 81 and 82 rigorously obeys detailed balance. For all equations in the body of the text, the state of the switching function is not shown, but is updated according to the scheme described. It should be noted, however that the ‘standard’ protocol is not updated according to this scheme, since there is no need to express the energies in terms of the latent states.

The acceptance probabilities for protocol B, as given in Eq. 62 are:

Equation 84

$$\left.\frac{acc_L(j\,|\,i)}{acc_L(i\,|\,j)}\right|_B = \frac{p_L(i,i,j)}{p_L(i,i,i)}$$

The ratio of the true acceptance probabilities is equivalent to the acceptance probabilities given in Eq. 60, and the ratio is given simply as:

Equation 85

$$\begin{aligned}
&\left(\left.\frac{acc(j\,|\,i)}{acc(j\,|\,i)}\right|_A\right)\Bigg/\left(\left.\frac{acc(j\,|\,i)}{acc(j\,|\,i)}\right|_B\right)\\
&= \frac{p_L(i,i,j)\,p_\varepsilon(i,j)}{p_L(i,i,i)}\cdot\frac{p_L(i,i,i)}{p_L(i,i,j)}\\
&= p_\varepsilon(i,j) = \exp[-\beta\varepsilon(i,j)]
\end{aligned}$$

Appendix B: Superposition of Multiple Histograms

The superposition of histograms from multiple simulations follows a procedure that weights the contribution of each histogram by the estimated error in each histogram at each bin site. This procedure has proven to give slightly better statistics when combining data from multiple simulations. We begin by generating an unnormalized histogram from an energy trajectory. We start by defining a vector of uniform bins:

Equation 86

$$\begin{gathered}
\mathbf{b} = \begin{bmatrix} E_{MIN} & E_{MIN}+\delta e & \ldots & E_{MAX}-\delta e & E_{MAX} \end{bmatrix}\\
b_j = E_{MIN} + j\delta e
\end{gathered}$$

where E_{MIN} and E_{MAX} are the minimum and maximum energies, respectively, and δe is the bin width, which, for the cases shown here, are equivalent for every bin. The unnormalized histogram of energies for a single energy trajectory is:

Equation 87

$$h_{ij} = \sum_t H(E_{i,t} > b_{i-1}) - H(E_{i,t} - b_i)$$

where the sum over *t* indicates that the entire (converged trajectory) is evaluated at each timestep. Here we identify the trajectory as the *i*th trajectory. H(x) is the Heaviside step function, and the convention $H(E_t > b_{i-1})$ indicates that the function has a value of unitiy when greater than b_{i-1}. This is simply an expression of the standard method for counting histograms. The normalized histogram is simply:

Equation 88

$$\hat{h}_{ij} = \frac{h_{ij}}{\sum_j h_{ij} \delta e}$$

A key to estimating the error in the estimate of the bin size is to assume that the variance is proportional to the number of entries in the bin[84]:

Equation 89

$$\sigma_{ij} = h_{ij}$$

If we use the standard error expression, we arrive at the following estimate of the error at each bin site:

Equation 90

$${\varepsilon_{ij}}^2 = \frac{{\sigma_{ij}}^2}{h_{ij} / g_i} = g_i h_{ij}$$

where the indices *i* and *j* refer to the simulation and bin number, respectively. Once this estimate is established, we simply use this idea to count the number of uncorrelated entries to the overall histogram in order to estimate the error. For a set of simulations I, we can generate a composite histogram $h_I^{(C)}$:

Equation 91

$$h_{I,j}^{(C)} = \sum_{i \in I} \frac{h_{ij}}{g_i}$$
$$\hat{h}_{I,j}^{(C)} = \frac{h_{I,j}^{(C)}}{\sum_j h_{I,j}^{(C)} \delta e}$$

with the error computed as:

Equation 92

$$\varepsilon_{I,j}^{(C)} = \sum_{i \in I} \frac{\varepsilon_{ij}}{g_i}$$
$$\hat{\varepsilon}_{I,j}^{(C)} = \frac{\varepsilon_{I,j}^{(C)}}{\sum_j h_{I,j}^{(C)} \delta e}$$

The main advantage to this method over a simple superposition of histograms is that it uses the estimate of the correlation interval for each trajectory in the weighting of the histograms. This approach naturally identifies the trajectories for which kinetic trapping is causing anomalously large correlation times. The error in the estimate of a correlation time is g +/- τ, so that an estimate of the correlation time can vary substantially from trajectory to trajectory.

Chapter 3

Backbone Sampling: Loop Closure Algorithms

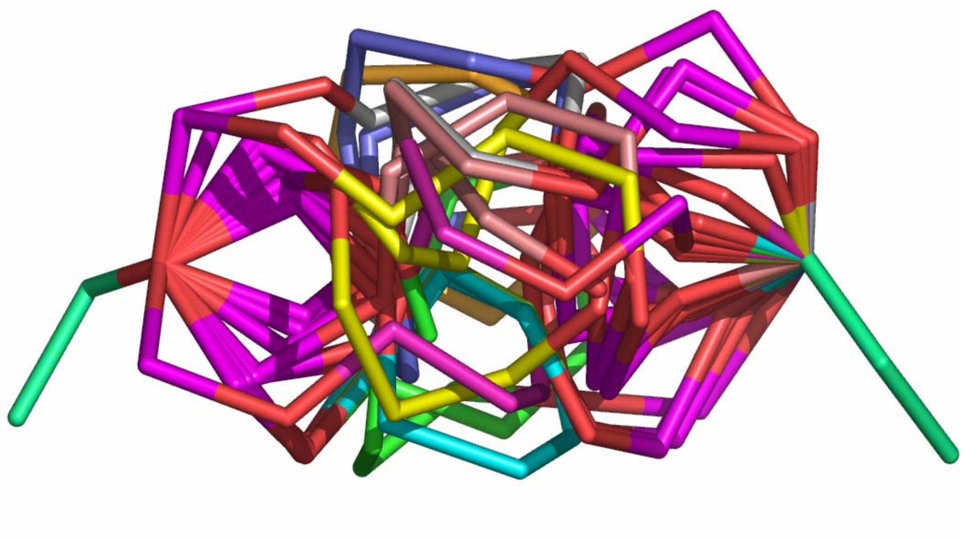


Figure 3.1 – A set of closed loops for an RNA backbone generated using the closure algorithms described. The endpoints are held at a fixed position, and the bond lengths and angles are also held fixed. The loop closure problem solves the system of equations which give the set of dihedral angles that satisfy these constraints.

Introduction

The idea of loop closure as applied to peptide systems was first introduced by Scheraga[85,86], and applied to systems such as cyclic peptides. Figure 3.1 shows a simple ensemble of closed loops for the backbone of an RNA structure. Here we can see that while the endpoints are held fixed, a wide variety of structures are present which satisfy the loop closure constraint. Having the capacity to explore alternative configurations of a closed loop can allow for the study of a wide variety of problems in protein sampling that are otherwise completely intractable, since the transitions for these types of motions would otherwise be of an inordinately long timescale. As shall be shown in later chapters, it is often the case that the fluctuations of a single loop in a protein can be the central feature contributing to its function.

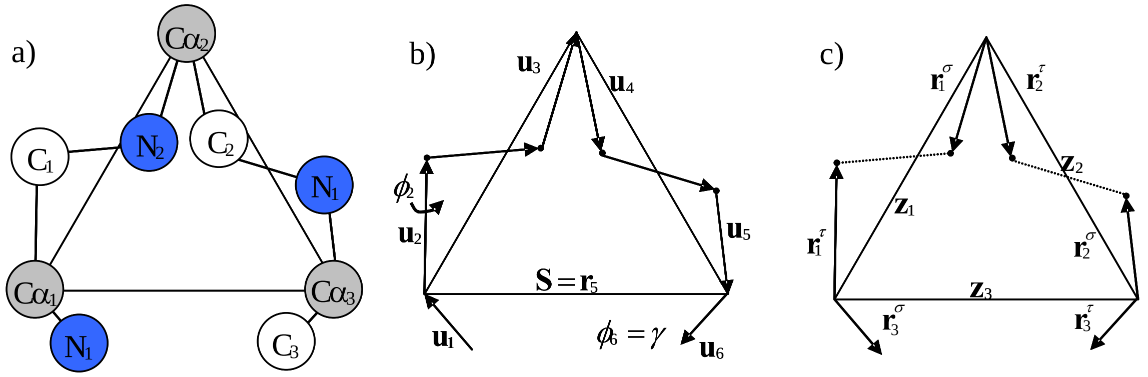


Figure 3.2 – Loop Closure coordinates and definitions. a) Coordinate labels for the tripeptide closure b) Scheraga coordinates with unit bond vectors c) Coutsias coordinate with bond vectors **r**. Notice that the vectors **z** form a closed triangle in a plane.

The goal then is to develop a way of locally sampling loop configurations while holding the remainder of the protein fixed. To accomplish this task, a variety of geometric algorithms are available, but the present work will focus on the original work of Scheraga and related implementations of Dinner[87], and the closely related work of Coutsias *et al*[88], as these algorithms are straightforward to implement in a way that allows for uniform sampling of dihedral coordinates. Both of the closure relations presented have an associated Jacobian determinant that can be computed directly.

The loop closure equations presented here are only a small subset of a very large body of work related to protein modeling and other fields of robotics and kinematics. Even the small sampling of equations here are only discussed to help to better understand the nature of the implementations involved. Much of the fundamental work in this area was conducted with Prof. Evangelos Coutsias, who very generously provided working loop closure codes that could be interfaced directly to the routines in the Protein Local Optimization Program (PLOP).

Loop Closure Equations

Scheraga Closure

The Scheraga closure is presented here primarily to motivate the development of the Jacobian determinant. Referring to Figure 3.2, and following Dinner's notation, we define each bond vector in the following way:

Equation 93

$$\mathbf{r}_{i-\mathbf{1}} = \mathbf{T}_{i-1}\mathbf{R}_i\mathbf{r}_i + \mathbf{p}_{i-\mathbf{1}}$$

where $\mathbf{r}_i$ is the bond vector pointing in the same direction of $\mathbf{u}_i$, $\mathbf{p}_i$ points to the origin of the *i*th coordinate system, and direction and **T** and **R** are rotation matrices, given by:

Equation 94

$$\mathbf{T}_{i-1} = \begin{pmatrix} \cos\theta_{i-1} & -\sin\theta_{i-1} & 0 \\ \sin\theta_{i-1} & \cos\theta_{i-1} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$\mathbf{R}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_{i-1} & -\sin\phi_{i-1} \\ 0 & \sin\phi_{i-1} & \cos\phi_{i-1} \end{pmatrix}$$

where θ and ϕ are the bond angles and dihedrals, respectively. This notation is somewhat standard, based on Flory's original descriptions. For the purposes of this work, the ω angles are held fixed, and so the transformation from $\mathbf{u}_2$ to $\mathbf{u}_3$ (for example), has an additional rigid body transformation (comprised of a 2 rotations and a displacement), which is implied, but omitted for simplicity. The closure equation that results is simply a statement of the following constraint:

Equation 95

$$\mathbf{S} = \mathbf{p}_0 + \sum_{I=1}^{5} \prod_{i=1}^{I} \mathbf{T}_{i-1}\mathbf{R}_i\mathbf{p}_i$$

with the additional constraints that place the coordinate system:

Equation 96

$$\mathbf{u}_6 = \prod_{i=1}^{6} \mathbf{T}_{i-1}\mathbf{R}_i\mathbf{e}_1$$

Equation 97

$$\mathbf{v} = \prod_{i=1}^{6} \mathbf{T}_{i-1}\mathbf{R}_i\mathbf{e}_2$$

where $\mathbf{e}_1 = (1,0,0)^T$ and $\mathbf{e}_2 = (0,1,0)^T$. Since $\mathbf{u}_6$ is a unit vector, there are only 2 degrees of freedom specified in Eq. 96. The direction of **v** relative to $\mathbf{u}_6$ gives the angle $\gamma = \phi_{6,}$ so that, when the 3 degrees of freedom specified by the constraint **S** are counted, a total of 6 degrees of freedom are specified. The bond angles and lengths are held fixed, such that the only unknown degrees of freedom are the dihedral degrees of freedom as given by the rotation matrices **R**. There are a total of 6 unknown dihedrals, and so a system of equations is now fully specified.

The solution of these equations is nontrivial, and not presented here. A salient feature of the system of equations is pointed out, however. There exists the possibility of multiple solutions for any given system of equations, and it is a requirement of the algorithm to be able to generate the full set of solutions for the system of equations. This is accomplished by a variety of trigonometric transformations, which will ultimately lead to a polynomial of 16th degree[86], permitting the use of Sturm's method to locate the full set of numerical roots to a polynomial equation.

Coutsias (CSJD) Closure

The Coutsias closure[88], developed by a collaborator, is presented here as well. The variables of interest are presented here to motivate the Jacobian formulation. The definitions here follow a slightly different convention, so we shall define the variables more carefully. This formulation follows the CSJD paper.

The vectors of interest are defined as follows:

Equation 98

$$
\begin{aligned}
\mathbf{z}_i &= \overrightarrow{C\alpha_i C\alpha_{i+1}} \\
\mathbf{r}_i^{\sigma} &= \overrightarrow{C\alpha_i C_i} \\
\mathbf{r}_i^{\tau} &= \overrightarrow{C\alpha_{i+1} N_{i+1}}
\end{aligned}
$$

and a simple coordinate system can be defined using the planar triangle formed by the closed loop of **z** vectors:

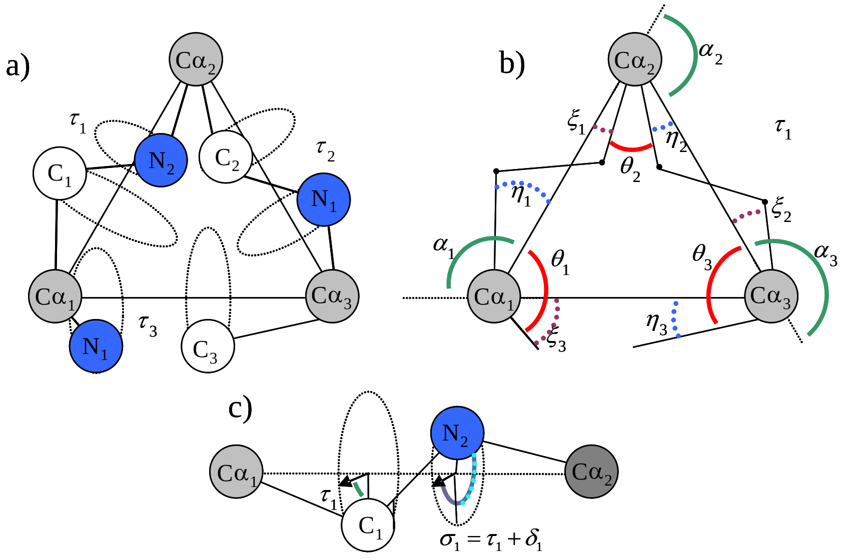


Figure 3.3 – Internal coordinates for CSJD closure (Figure 3.adapted from Coutsias *et al*[88]) a) dihedral angles τ which are the unknown variables, b) Internal variables definitions c) τ and σ are directly related.

Equation 99

$$\hat{\mathbf{y}} = \frac{\mathbf{z_3} \times \mathbf{z_1}}{|\mathbf{z_3} \times \mathbf{z_1}|}$$
$$\hat{\mathbf{x}}_i = \hat{\mathbf{y}} \times \hat{\mathbf{z}}_i$$

The angles are defined as follows:

Equation 100

$$\begin{aligned} \alpha_i &= \cos^{-1}(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_{i-1}) \\ \eta_i &= \cos^{-1}(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{r}}_i^{\tau}) \\ \xi_i &= \cos^{-1}(\hat{\mathbf{z}}_i \cdot \hat{\mathbf{r}}_i^{\sigma}) \\ \tau_i &= \sigma_i + \delta_i \end{aligned}$$

The vectors pointing from the α carbons to the adjacent atoms can be defined in terms of the internal coordinate system:

Equation 101

$$\begin{aligned} \hat{\mathbf{r}}_i^{\tau} &= \cos\eta_i \hat{\mathbf{z}}_i + \sin\eta_i(\cos\tau_i \hat{\mathbf{x}}_i + \sin\tau_i \hat{\mathbf{y}}) \\ \hat{\mathbf{r}}_{i-1}^{\sigma} &= \cos\xi_{i-1} \hat{\mathbf{z}}_{i-1} + \sin\xi_{i-1}(\cos\sigma_{i-1} \hat{\mathbf{x}}_{i-1} + \sin\sigma_{i-1} \hat{\mathbf{y}}) \end{aligned}$$

The constraints can now be enforced by imposing the rigidity of the angles at the nodes:

Equation 102

$$\hat{\mathbf{r}}_i^{\tau} \cdot \hat{\mathbf{r}}_{i-1}^{\sigma} = \cos\theta_i$$

resulting in a system of 3 equations and 3 unknowns (τ_1, τ_2, τ_3). Further rearrangement of the equations is detailed in Coutsias *et al*, but it is noteworthy that the resulting equation is also a polynomial of 16th order, which is amenable to the same solution method as the Scheraga closure (although a more sophisticated method is used here). The closure equations always produce an even number of solutions, and the solutions are equivalent using either the Scheraga or Coutsias method. It is also noteworthy that the formulation presented here connects to a large body of kinematics literature, and future work may expand on these formulations for more specific loop closures.

The Jacobian coordinate transformation

Now that the variables have been defined for each of the formalisms, is of interest to derive the Jacobian in both coordinate systems. The Jacobian coordinate transformation allows us to express the change in differential volume (or hypervolume) when transforming from one coordinate system to another. Expressing the transformation from coordinates **a** to **b**, we can write:

Equation 103

$$d\mathbf{a} = J\left(\frac{\mathbf{a}}{\mathbf{b}}\right)d\mathbf{b}$$

Where d**a**=$da_1da_2\ldots da_N$ and d**b**=$da_1da_2\ldots da_N$. This is often thought of as a chain rule in higher dimensions. The *Jacobian* is the determinant of the *Jacobi matrix*:

Equation 104

$$J\left(\frac{\mathbf{a}}{\mathbf{b}}\right) = \det\left(\frac{\partial(\mathbf{a})}{\partial(\mathbf{b})}\right)$$

The Jacobi matrix is a square matrix, sometimes notated as:

Equation 105

$$\frac{\partial(\mathbf{a})}{\partial(\mathbf{b})} = \frac{\partial(a_1, a_1 \ldots a_N)}{\partial(b_1, b_1 \ldots b_N)}$$

and the elements are are given by:

Equation 106

$$\left.\frac{\partial(\mathbf{a})}{\partial(\mathbf{b})}\right|_{ij} = \frac{\partial a_i}{\partial b_j}$$

Jacobian in Flory-Scheraga Coordinates

The Jacobian required by the concerted rotation in the Flory-Scheraga coordinate system comes from the following transformation:

Equation 107

$$d\mathbf{r}_5 d\mathbf{u}_6 d\gamma = J\left(\frac{\boldsymbol{\varphi}}{\mathbf{r}_5, \mathbf{u}_6, \gamma}\right) d\boldsymbol{\varphi}$$

Where the coordinates shown are given in Figure 3.2a. The reason for the transformation here is due to the fact that the dihedral space would not be sampled randomly due to the loop constraints.

The Jacobian shown is not easily calculated, but the inverse relation is more straightforward to compute. This was first calculated by Dodd[89]. Here we present is in contrast to the Jacobian calculated in SCJD coordinates. The inverse Jacobian is related in the following way:

Equation 108

$$J\left(\frac{\boldsymbol{\varphi}}{\mathbf{r}_5, \mathbf{u}_6, \gamma}\right) = 1/J\left(\frac{\mathbf{r}_5, \mathbf{u}_6, \gamma}{\boldsymbol{\varphi}}\right)$$

where

Equation 109

$$\frac{\partial(\mathbf{r}_5,\mathbf{u}_6,\gamma)}{\partial(\boldsymbol{\varphi})}=\begin{pmatrix} \frac{\partial\mathbf{r}_5}{\partial\varphi_1} & \frac{\partial\mathbf{r}_5}{\partial\varphi_2} & \frac{\partial\mathbf{r}_5}{\partial\varphi_3} & \frac{\partial\mathbf{r}_5}{\partial\varphi_4} & \frac{\partial\mathbf{r}_5}{\partial\varphi_5} & \frac{\partial\mathbf{r}_5}{\partial\varphi_6} \\ \frac{\partial\mathbf{u}_6}{\partial\varphi_1}\cdot\mathbf{e}_1 & \frac{\partial\mathbf{u}_6}{\partial\varphi_2}\cdot\mathbf{e}_1 & \frac{\partial\mathbf{u}_6}{\partial\varphi_3}\cdot\mathbf{e}_1 & \frac{\partial\mathbf{u}_6}{\partial\varphi_4}\cdot\mathbf{e}_1 & \frac{\partial\mathbf{u}_6}{\partial\varphi_5}\cdot\mathbf{e}_1 & \frac{\partial\mathbf{u}_6}{\partial\varphi_6}\cdot\mathbf{e}_1 \\ \frac{\partial\mathbf{u}_6}{\partial\varphi_1}\cdot\mathbf{e}_2 & \frac{\partial\mathbf{u}_6}{\partial\varphi_2}\cdot\mathbf{e}_2 & \frac{\partial\mathbf{u}_6}{\partial\varphi_3}\cdot\mathbf{e}_2 & \frac{\partial\mathbf{u}_6}{\partial\varphi_4}\cdot\mathbf{e}_2 & \frac{\partial\mathbf{u}_6}{\partial\varphi_5}\cdot\mathbf{e}_2 & \frac{\partial\mathbf{u}_6}{\partial\varphi_6}\cdot\mathbf{e}_2 \\ \frac{\partial\gamma}{\partial\varphi_1} & \frac{\partial\gamma}{\partial\varphi_2} & \frac{\partial\gamma}{\partial\varphi_3} & \frac{\partial\gamma}{\partial\varphi_4} & \frac{\partial\gamma}{\partial\varphi_5} & \frac{\partial\gamma}{\partial\varphi_6} \end{pmatrix}$$

which is a 6x6 square matrix. With some rearrangements, a 5x5 matrix is obtained, and can be expressed in terms of the cross products of vectors:

Equation 110

$$\frac{\partial(\mathbf{r}_5,\mathbf{u}_6,\gamma)}{\partial(\boldsymbol{\varphi})}=\begin{pmatrix} \mathbf{u}_1\times(\mathbf{r}_5-\mathbf{r}_1) & \mathbf{u}_1\times(\mathbf{r}_5-\mathbf{r}_2) & \mathbf{u}_1\times(\mathbf{r}_5-\mathbf{r}_3) & \mathbf{u}_1\times(\mathbf{r}_5-\mathbf{r}_4) & 0 \\ (\mathbf{u}_1\times\mathbf{u}_6)\cdot\mathbf{e}_1 & (\mathbf{u}_2\times\mathbf{u}_6)\cdot\mathbf{e}_1 & (\mathbf{u}_3\times\mathbf{u}_6)\cdot\mathbf{e}_1 & (\mathbf{u}_4\times\mathbf{u}_6)\cdot\mathbf{e}_1 & (\mathbf{u}_5\times\mathbf{u}_6)\cdot\mathbf{e}_1 \\ (\mathbf{u}_1\times\mathbf{u}_6)\cdot\mathbf{e}_2 & (\mathbf{u}_2\times\mathbf{u}_6)\cdot\mathbf{e}_2 & (\mathbf{u}_3\times\mathbf{u}_6)\cdot\mathbf{e}_2 & (\mathbf{u}_4\times\mathbf{u}_6)\cdot\mathbf{e}_2 & (\mathbf{u}_5\times\mathbf{u}_6)\cdot\mathbf{e}_2 \end{pmatrix}$$

The details of how these matrices are obtained are well documented, and we present them here in comparison with the CSJD Jacobian.

Jacobian in CSJD Coordinates

The Coutsias closure gives a system of equations with 3 unknowns, rather than a system of equations with 6 unknowns. It stands to reason then that a simpler Jacobian would be possible to compute. To begin, the local conformer (backbone) coordinates can be written as:

Equation 111

$$d\{\mathbf{r}^{<C,Local>}\}=d\mathbf{r}_1^{\tau}d\mathbf{m}_1(-d\mathbf{r}_1^{\sigma})d\mathbf{r}_2^{\tau}d\mathbf{m}_2(-d\mathbf{r}_2^{\sigma})d\mathbf{r}_3^{\tau}d\mathbf{m}_3(-d\mathbf{r}_3^{\sigma})$$

where we shall adopt a local coordinate system of the loop as our basis (say, coordinate system 3). The vectors **m** are the bond vectors connecting the carbonyl carbon and nitrogen groups. Each triplet of variables is dependent, and can be reduced to a single vector variable. For example, the bond angle constraint given by:

Equation 112

$$d\hat{\mathbf{r}}_i^{\tau} \cdot d\hat{\mathbf{r}}_{i-1}^{\sigma} = \cos(\theta_i)$$

Implies that a series of rigid rotations will transform all $\mathbf{r}^{\sigma}$ to $\mathbf{r}^{\tau}$, with a Jacobian of unity. Since the (fixed) dihedral rotation is not given in SCJD, we will not express the rotations explicitly, in order to avoid extra variables. The second angular axis of rotation can be chosen from the internal coordinates of the alpha carbon. Likewise, the constraint on δ_i implies that the **m** vectors can be generated by similar rigid rotations about the bond angle and δ_i. These rigid rotations constrain the extra variables to fall along the path of integration of the 3 independent variables. The three independent vector variables are:

Equation 113

$$d\mathbf{r}_1^{\tau} d\mathbf{r}_2^{\tau} d\mathbf{r}_3^{\tau}$$

We could have easily chosen the sigma vectors as the independent variables, which would ultimately lead to the same Jacobian. This is analogous to the starting point of the Scheraga closure (Eq. 107). Another way of expressing differentials of this type is:

Equation 114

$$d\mathbf{r}_1^{\tau} d\mathbf{r}_2^{\tau} d\mathbf{r}_3^{\tau} = \left(d\mathbf{r}_1^{\tau} \times d\mathbf{r}_2^{\tau}\right) \cdot d\mathbf{r}_3^{\tau} = \left[\left(\frac{\partial \mathbf{r}_1^{\tau}}{\partial \tau_1} d\tau_1\right) \times \left(\frac{\partial \mathbf{r}_2^{\tau}}{\partial \tau_2} d\tau_2\right)\right] \cdot \frac{\partial \mathbf{r}_3^{\tau}}{\partial \tau_2} d\tau_2$$

Equation 115

$$= \left[\left(\frac{\partial \mathbf{r}_1^{\tau}}{\partial \tau_1}\right) \times \left(\frac{\partial \mathbf{r}_2^{\tau}}{\partial \tau_2}\right)\right] \cdot \frac{\partial \mathbf{r}_2^{\tau}}{\partial \tau_2} d\boldsymbol{\tau} = \left|\left[\frac{\partial \mathbf{r}_1^{\tau}}{\partial \tau_1} \quad \frac{\partial \mathbf{r}_2^{\tau}}{\partial \tau_2} \quad \frac{\partial \mathbf{r}_3^{\tau}}{\partial \tau_2}\right]\right| d\boldsymbol{\tau}$$

where the 3x3 matrix in square brackets is the effective Jacobi matrix. These vectors must be expressed in the same basis set. It is worthwhile to restate the definition of $\mathbf{r}^{\tau}$ in SCJD:

Equation 116

$$\begin{aligned}\hat{\mathbf{r}}_i^{\tau} &= \cos\eta_i\hat{\mathbf{z}}_i + \sin(\eta_i)(\cos\tau_i\hat{\mathbf{x}}_i + \sin\tau_i\hat{\mathbf{y}}_i) \\ \hat{\mathbf{r}}_i^{\tau} &= \left[\sin\eta_i\cos\tau_i \quad \sin\eta_i\sin\tau_i \quad \cos\eta_i\right]^T_{<i>}\end{aligned}$$

Where we have added the subscript <*i*> to indicate which coordinate set basis is being used, since there are three basis sets used in the loop. The relation of one basis set to another is given by $\mathbf{b}_{i+1}=\mathbf{C}_{i+1}\mathbf{b}_i$, where:

Equation 117

$$\mathbf{C}_i = \begin{bmatrix} \cos\alpha_i & 0 & -\sin\alpha_i \\ 0 & 1 & 0 \\ \sin\alpha_i & 0 & \cos\alpha_i \end{bmatrix}$$

It is convenient to express the partial derivatives in the following notation:

Equation 118

$$(\delta\mathbf{r}_i)\big|_{<i>} = \frac{\partial\mathbf{r}_i^{\tau}}{\partial\tau_i} = \left[-\sin\eta_i\sin\tau_i \quad \sin\eta_i\cos\tau_i \quad 0\right]^T_{<i>}$$

where the subscript <*i*> again reminds us of which basis we are expressing the vector. We now express the effective Jacobi matrix compactly as:

Equation 119

$$\frac{\partial(\mathbf{r}^{\sigma})}{\partial(\boldsymbol{\tau})} = \left[(\delta\mathbf{r}_1)\big|_{<3>} \quad -(\delta\mathbf{r}_2)\big|_{<3>} \quad -(\delta\mathbf{r}_3)\big|_{<3>}\right]$$

Where each column has 3 elements. Here, we are using 3rd basis set, since $\mathbf{z}_3$ is held fixed as part of the loop closure requirement. Notice, however, that the 3rd basis does not remain fixed. We can express the Jacobi matrix in terms of the native bases as:

Equation 120

$$\frac{\partial(\mathbf{r}^{\sigma})}{\partial(\boldsymbol{\tau})} = \left[\mathbf{C}_1(\delta\mathbf{r}_1)\big|_{<1>} \quad \mathbf{C}_2\mathbf{C}_1(\delta\mathbf{r}_2)\big|_{<2>} \quad (\delta\mathbf{r}_3)\big|_{<3>}\right]$$

We can now express each column explicitly in terms of the local angular coordinates:

Equation 121

$$\mathbf{C}_1(\delta\mathbf{r}_1)\big|_{<1>} = \sin\eta_1 \begin{bmatrix} -\cos\alpha_1 \sin\tau_1 \\ \cos\tau_1 \\ -\cos\alpha_1 \sin\tau_1 \end{bmatrix}$$

Equation 122

$$\mathbf{C}_2\mathbf{C}_1(\delta\mathbf{r}_2)\big|_{<2>} = \sin\eta_2 \begin{bmatrix} -(\cos\alpha_2\cos\alpha_1 - \cos\alpha_2\cos\alpha_1)\sin\tau_2 \\ \cos\tau_2 \\ -(\sin\alpha_2\cos\alpha_1 + \cos\alpha_2\sin\alpha_1)\sin\tau_2 \end{bmatrix}$$

Equation 123

$$(\delta\mathbf{r}_3)\big|_{<3>} = \sin\eta_3 \begin{bmatrix} -\sin\tau_3 \\ \cos\tau_3 \\ 0 \end{bmatrix}$$

It is not necessary to express this matrix in angular coordinates, although it is believed to be more efficient. If we wish to express the Jacobian in terms of Cartesian components, as does Theodorou and others, we can write:

Equation 124

$$\frac{\partial(\mathbf{r}^{\tau})}{\partial(\boldsymbol{\tau})} = \left[\mathbf{z}_1 \times \mathbf{r}_1^{\tau} \quad \mathbf{z}_2 \times \mathbf{r}_2^{\tau} \quad \mathbf{z}_3 \times \mathbf{r}_3^{\tau}\right]$$

In practice, these Jacobians should be equivalent, and example calculations have shown this to be the case, but the original Dodd and Theodorou Jacobian is in the current implementation. The novel calculation of the Jacobian in CSJD is presented here for completeness and general interest.

Implementation Details

Since the loop closure algorithm works only for 6 dihedrals, additional logic is added to extend this functionality to loops of arbitrary length. Figure 4.5 illustrates how this is implemented. As is the case with all previous descriptions, the ω angles are held fixed to the native values. Each a carbon is treated as a node.

To begin, a ϕ or ψ angle is chosen at random and perturbed uniformly. This is referred to as the *driver angle.* A triangle connecting α carbons is constructed randomly about this perturbation, and the triangle that is constructed is treated as a closed loop, with only the ϕ/ψ pairs adjacent to the α carbons selected allowed to move, with the remaining portions of the loop treated as rigid bodies. This procedure requires a loop with at least 4 α carbons, so that one α carbon node contains the driver angle, and the remaining 3 α carbon nodes can be used to construct the loop. A stationary set of solutions is also generated, for which the change in the driver angle is zero.

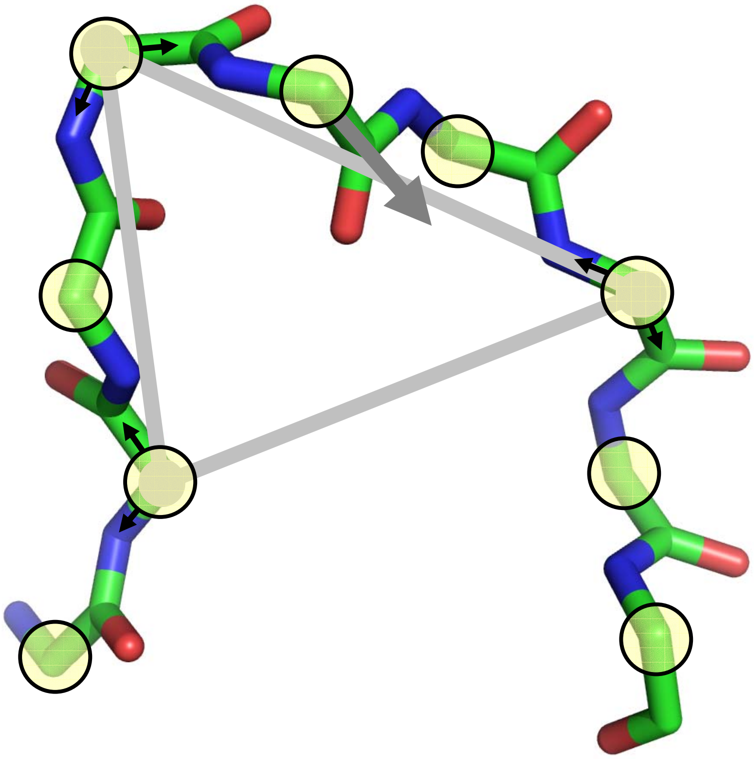


Figure 3.4 – Loop Closure for a loop of arbitrary length. The dark gray arrow points along the randomly selected driver angle dihedral, and the light gray triangle shows a randomly constructed triangle around the driver angle. The 6 free dihedrals (black arrows) are constructed to be directly adjacent to the Cα nodes forming the triangle.

This procedure generates a minimum of two and a maximum of 32 solutions. A solution is selected randomly from this set with the following probability:

Equation 125

$$\alpha(\boldsymbol{\varphi} \rightarrow \boldsymbol{\varphi'}) = \frac{J^{S}(\boldsymbol{\varphi'})}{\sum_{i=1}^{N_S^{(\Delta\phi_D=0)}} J(\boldsymbol{\varphi}_i^{(\Delta\phi_D=0)}) + \sum_{i=1}^{N_S^{(\Delta\phi_D=\xi)}} J(\boldsymbol{\varphi}_i^{(\Delta\phi_D=\xi)})}$$

where $J^{S}(\boldsymbol{\varphi'})$ is the set of dihedrals selected from the ensemble of solutions to the loop closure equations, $J(\boldsymbol{\varphi}_i^{(\Delta\phi_D=0)})$ is a member of the stationary ensemble of solutions, for which the number of solutions is $N_S^{(\Delta\phi_D=0)}$. The set of solutions associated for which the driver angle is assigned a uniform variate ξ is $J(\boldsymbol{\varphi}_i^{(\Delta\phi_D=\xi)})$, with the number of solutions being $N_S^{(\Delta\phi_D=\xi)}$. The minimum number of stationary solutions is 2, 1 of which is the 'self solution' or, unmodified coordinate state. The number of solutions is always even, due to the symmetry of constructing a loop. The minimum number of solutions with the perturbed driver angle can be zero, as the driver angle can generate a set of dihedrals for which there is no solution. Overall, there is a guarantee of at least 2 solutions to choose from every time a loop closure move is generated.

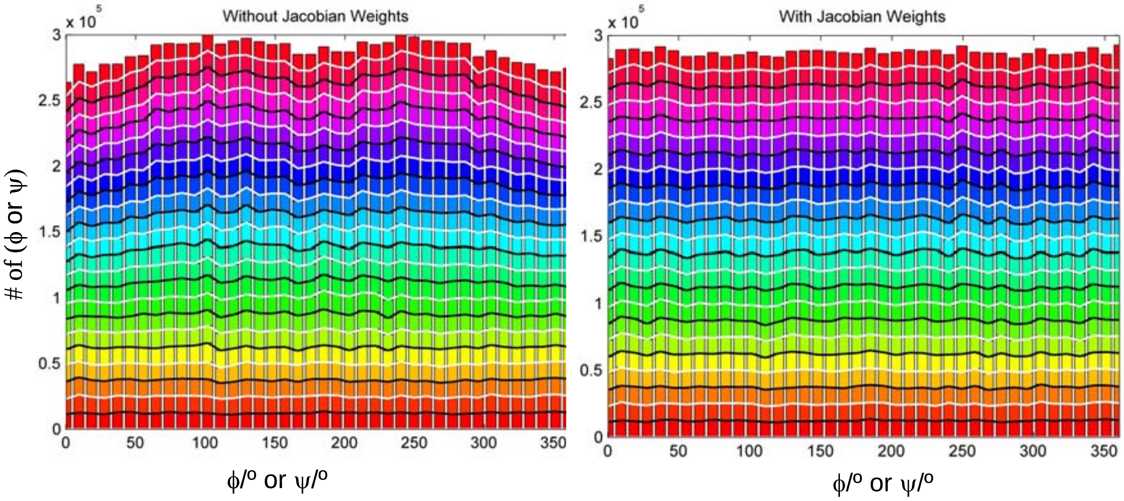


Figure 3.5 – Distribution of ϕ/ψ angles with and without Jacobian weighting of selection for an 11 residue peptide. No forcefield is used in the selection or acceptance probability.

Figure 3.5 shows a stacked histogram of ϕ,ψ angles sampled using a 50% mixture of uniformly perturbed dihedrals and loop closure moves. This closely follows previous methods to demonstrate the uniformity[87,89], and a similar approach is used here. The free dihedral perturbation is needed in order to allow the loop closure moves to sample over the ensemble of all possible loop closure configurations, as an ensemble of dihedrals for a single closed loop closure will have geometrically occluded regions of space. The motivation for showing the non Jacobian sampling is to show that the introduction of the free dihedrals does not guarantee uniformity, and that the use of the Jacobian generates a uniform distribution.

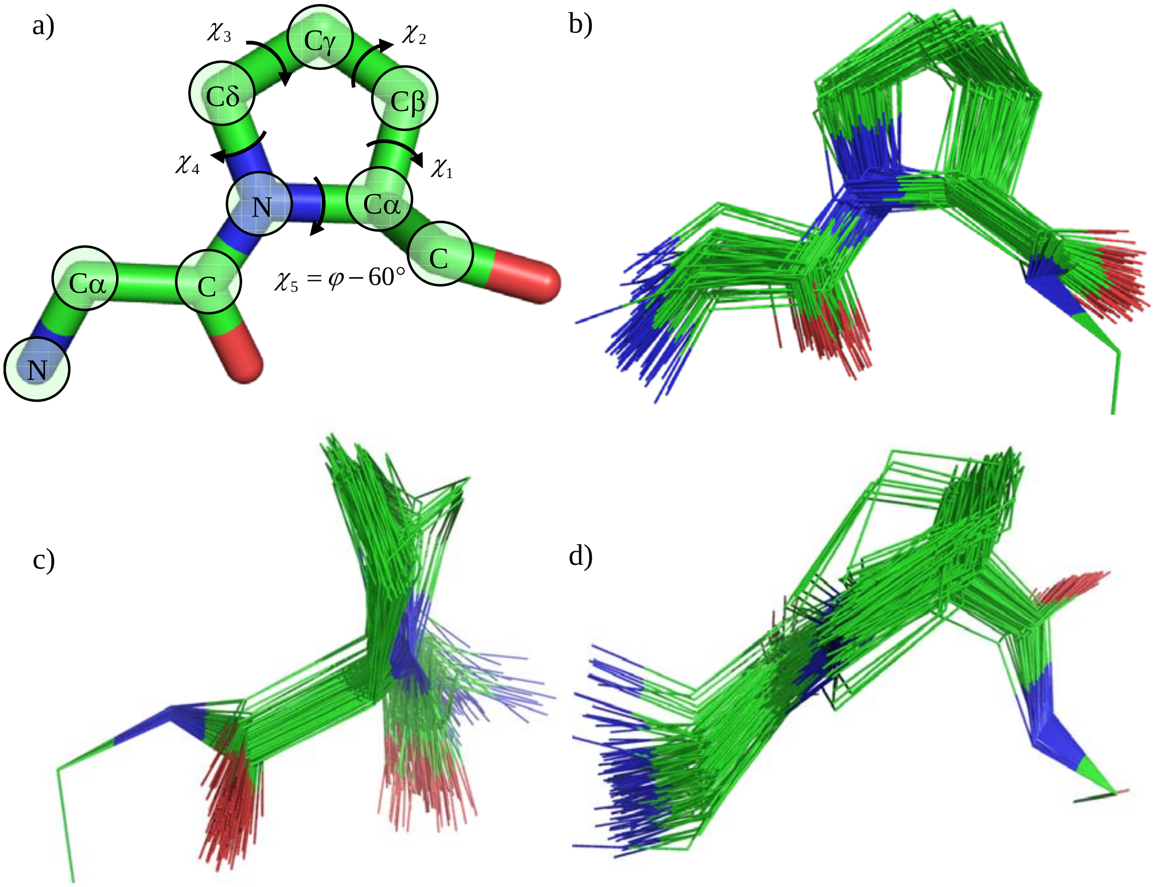


Figure 3.6 – Proline Loop Closure a) Loop closure variables (Figure 3.adapted from Ho *et al*[90]) b) Example of an ensemble of proline states and related backbone angles c) Side view showing alternative pucker states d) top view showing alternative pucker states.

Specialized Closures - Proline Pucker

The proline ring is a cyclic five member ring that is constructed about the Cα-N peptide bond, which directly couples the backbone angle[91]. The closure equations are presented in Ho *et al,* and Figure 3.1a shows the dihedral coordinates of interest for this system. There are 3 properties of this closure that are of relevance to the design. The first is the fact that the loop closure equations permit up to 2 solutions, which represent what is typically understood to be 'pucker up' and 'pucker down' solutions. The second is that Cβ– Cγ– Cδ angle is allowed to fluctuate to accommodate the strained geometry of the ring, which deviates from detailed balance in the sense that the bond angles should be held fixed. The third is that any perturbation to the backbone angle of a proline must be screened in some way to ensure a closed proline loop is feasible to construct.

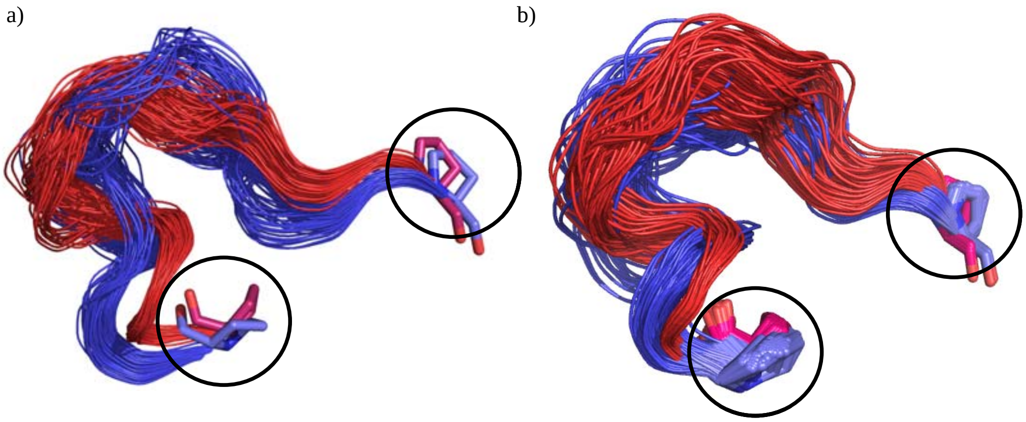


Figure 3.7 – Effect of proline sampling on accessible configurations of TIM a) TIM with no proline closure moves closed form is in blue and open form is in red. b) Same simulation conditions, but with the proline closure added.

So, for a sidechain perturbation of proline, (where the backbone is not necessarily perturbed), the sampling consists of solving the closure and selecting between the pucker states with equal probability. For any loop closure move, the move is first screened to see whether the ϕ angle falls within the allowable range, and then a closure move is generated with the new perturbed angle, followed by a selection of pucker state as described above. No Jacobian is computed for this closure. Typically, the pucker selection is very strongly enthalpically driven, and so the deviation from detailed balance in these cases is not even detectable as a source of error. Figure 3.7 shows an example trajectory of the loop of TIM, which is flanked by prolines. These effects are particularly pronounced for Monte Carlo trajectories. Notice that even harmonic fluctuations are inaccessible, since any perturbation requires the correct geometric concerted move of coordinates.

Future Directions for Loop Closures

Loop closure algorithms and there extensions have made key aspects of Monte Carlo sampling accessible. Simple extensions to the current closure approach could include more sophisticated definitions of what a ‘loop’ is, which could include disulfide bridging, or virtual geometric networks (such as hydrogen bonding), whose properties are thought to be conserved. Of course, the extension to DNA and RNA is of interest, with a host of implementation issues to consider. The kinematics description also affords a wide variety of generalizations, including those which conserve geometries in new and more helpful ways. A simple example of this is to generate a loop closure move which preserves the location of a point along the loop in Cartesian space, and reconstructs alternative configurations according to this new constraint. As new systems and approaches emerge, optimal geometric algorithms will always play a central role in an optimal design.

Chapter 4

Monte Carlo sampling with hierarchical move sets: POSH Monte Carlo

Jerome Nilmeier

Graduate Group in Biophysics, University of California at San Francisco

Matt Jacobson

Department of Pharmaceutical Chemistry University of California at San Francisco

Abstract

We present a new Monte Carlo method for sampling rugged energy landscapes that allows for efficient transitions across sparsely distributed local basins. The trial move consists of two parts: the initial move consists of a large, coarse trial move, and the second part is a Monte Carlo trajectory generated using smaller trial moves. To maintain detailed balance, a reverse transition probability is estimated along a path that differs from the forward path. Since the forward and reverse transitions are different, we label the algorithm POSH (Port Out, Starboard Home) Monte Carlo. The process obeys detailed balance to the extent that the transition probabilities are correctly estimated. There is an optimal range of performance for a given energy landscape, which depends on how sparsely the low energy states of the system are distributed. For simple model systems, there is no upper bound to the number of inner steps. The phosphopeptide Ace-Gly-Ser-pSer-Ser-Nma is studied as a proof of principle for the algorithm in a biomolecular application. For the system studied, we show that POSH sampling

generates precise distributions using the number of inner steps set to up to 20. NMR observables also compare well with experimental values.

Introduction

The Metropolis algorithm[4] has been in use for over 50 years, with generalizations of the idea applied to fields far beyond the field of molecular modeling, for which it was initially developed. For systems with densely packed geometries, the generation of good trial moves can be a substantial challenge. A variety of methods in the field of liquid simulation have been developed to address this type of problem[92,93,94]. For polymeric systems, chain growth methods[11,13] and other methods using the idea of biased sampling [10,95,96] have become a staple of the field, as well as simple pivot moves[97]. The Rosenbluth methods have been applied in continuum applications as well[98,99].

For biomolecular systems, methods such as Monte Carlo Minimization[100] have emerged as a practical method for sampling landscapes that have multiple, sparsely distributed minima. This produces an approximately correct distribution, and is able to produce an ergodic distribution of configurations. More rigorous estimates basin entropy can be made, as is the case with the Mining Minima approach[101,102].

There are many sampling algorithms which are designed expressly for crossing large energetic barriers and obeying, or nearly obeying, detailed balance. Most of these involve generating a Markov chain of states (a walker) in an expanded state space[103], which can be accepted to the configuration space of interest based on a modified acceptance probability [104-108]. The most widely adopted of these types is the replica exchange method [109], and other closely related methods[110-113], which are particularly well suited to parallel processing. The success of these approaches has led to a variety of sampling methods that emphasize locating global minima. In many cases, particularly in difficult biomolecular optimization problems such as protein folding[2,3], drug design, and homology modeling[114-117], where the challenges of structure prediction are driven primarily by enthalpic and geometric considerations, the use of minimization and other optimization approaches have become the mainstay of the field, with entropic considerations added as a secondary effect.

In this work we propose a simple, general method for sampling rugged landscapes that obeys detailed balance. The basic idea is to generate a large initial trial perturbation, followed by a series of small Monte Carlo moves to anneal the initial trial move to a lower energy, and accept resulting trial move with a modified acceptance probability.

Motivation

Partitioning of configuration space

A Monte Carlo sampling strategy seeks to evaluate integrals of the type

Equation 126

$$
\begin{aligned}
< O > &= \frac{1}{Z}\int d\mathbf{q} O(\mathbf{q}) \exp(-\beta U(\mathbf{q})) \\
Z &= \int d\mathbf{q} \exp(-\beta U(\mathbf{q}))
\end{aligned}
$$

where **q** is the set of all coordinates of the system of interest, $\beta = (k_B T)^{-1}$ is the inverse temperature, and $U(\mathbf{q})$ is the potential energy. The observed quantity $\langle O \rangle$ is averaged over many instances $O(\mathbf{q})$. Z is the normalization constant, or configuration integral.

The Boltzmann factor in the integrand is the unnormalized probability distribution.

Equation 127

$$p(\mathbf{q}) = \exp(-\beta U(\mathbf{q}))$$

Often, a natural partition of the entire space becomes convenient. A common instance of this assertion is the Born-Oppenheimer approximation, where the nuclear degrees of freedom are considered to be uncoupled, or adiabatic, relative to the electronic degrees of freedom[13]. This type of approximation also appears in the formulation of the implicit solvation model, where the solvent degrees of freedom are integrated out, and an approximate model for the interaction between a macromolecule and the solvent (**y**) is introduced [19]. Propagation along adiabatic degrees of freedom has been introduced in both Monte Carlo[98,118,119] and Dynamical[118-120] contexts. For the application considered, the coordinate decomposition is between protein backbone and sidechain coordinates.

The motivation for sampling separate subspaces is often guided by the assertion that a partitioning of configuration space can be defined where the covariant fluctuations between the partitioned subspaces is small. This can often be justified by a dynamical argument, as is frequently the case for the examples given above. The practical motivation for decomposition of subspaces is often much more compelling, however. In proteins, for example, different geometric algorithms are appropriate for sampling backbone [86,88] and sidechain degrees of freedom[121], and the challenge lies in combining these trial moves in a way that preserves ergodicity, as well also generating a high acceptance ratio, and, of course, the expected distribution of states.

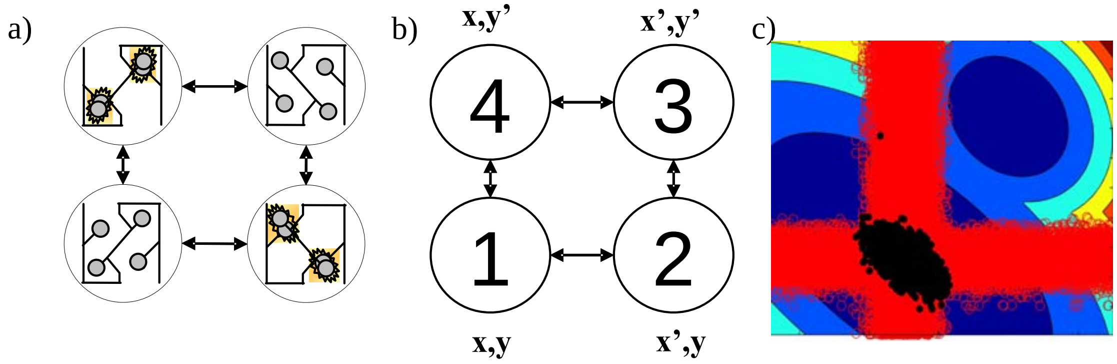


Figure 4.1 a) schematic of the multiple subspace sampling problem for proteins. Backbone and sidechain coordnates are **f** and **c**, respectively b) Generalization of the 2 subspace sampling problem. x and y represent subspaces to be sampled c) Monte Carlo sampling of a 2 dimensional landscape. Points in red are trial configurations, and points in black are accepted configurations.

Figure 4.1a shows a schematic to motivate the development of a sampling protocol. The complete configurational space is partitioned into torsional coordinates (ϕ) of the backbone and torsional coordinates of the sidechains (χ). Consider a trial move that consists of randomly selecting a subspace to perturb (ϕ or χ), and accepting with the Metropolis criterion. A single perturbation in backbone space may generate a configuration with the sidechains in a high energy state. Likewise, a perturbation of sidechains only may also lead to a high energy state. Although a series of samples from the trial state could generate a lower energy state, the initial move would be rejected outright with a standard scheme. If both degrees of freedom were perturbed simulaneously, however, the likelihood of generating a reasonable trial can become vanishingly small. The generalization of this problem to a topological model and a continuum representation is shown in Figure 4.1b. For the remainder of the work, the *original* coordinate state will be labeled as state 1, the *initial trial state* will be state 2 and the *final trial* state will be labeled state 3. The *reflected trial* state, labeled as state 4, will be defined shortly.

Figure 4.1c shows two basins separated by large energy barriers on a simple two dimensional energy landscape. The trial moves shown here are also unable to sample across basins effectively, due to the nature of the trial move set. A system that remains in a macrostate for long times relative to local correlations is often referred to as quasi-ergodic[105], frustrated/glassy[122], or kinetically trapped. In the cases shown for Figure 4.1, the most intuitive solution is to generate a series of Monte Carlo steps in the alternate subspace in order to locate a low energy coordinate, and then accept the final trial coordinate with some reasonable probability. In this sense, the transition the initial trial to the final trial can be thought of as an annealing step.

Theory

Detailed Balance

To determine the proper acceptance criterion, the condition of detailed balance is:

Equation 128

$$p_i T_{ij} = p_j T_{ji}$$

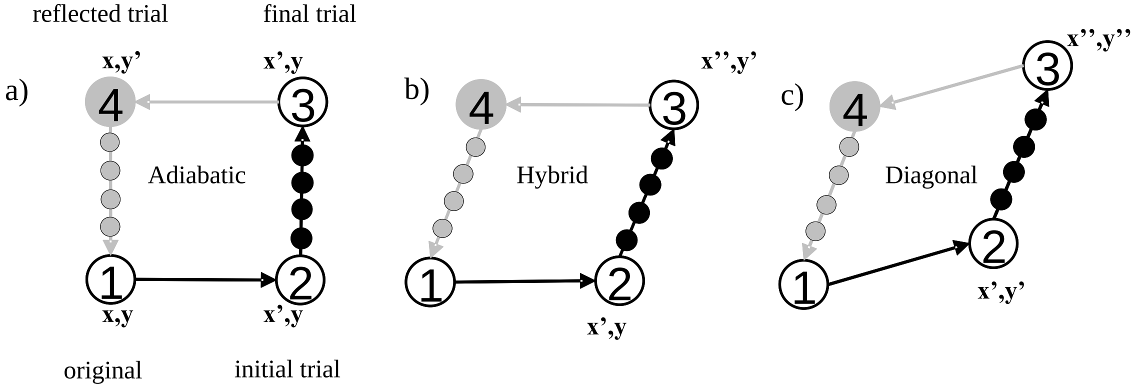


Figure 4.2 - POSH pathways a) Adiabatic pathway b) Hybrid Pathway c) Diagonal Pathway. Calculation of coordinate 4 is described in the text, and in Figure 4.3. Coordinate labels in 4.2a correspond to state numbers, and are equivalent in figures b and c.

where i and j are two arbitrary coordinate states with probabilities $p_i = p(\mathbf{q}_i)$ and $p_j = p(\mathbf{q}_j)$, as given by Eq. 127, respectively. The transition probability T_{ij} is the probability of transitioning to coordinate state j from state i, and T_{ji} is the reverse transition probability. The condition of detailed balance will be applied to the states 1 and 3, as shown in Figure 4.2:

Equation 129

$$p_1 T_{13} = p_3 T_{31}$$

where the forward transition consists of a trial move followed by a chain of moves. In the cases described by Figure 4.2a, for example, the initial trial could be also notated as $p_2 = P(\mathbf{x'}, \mathbf{y})$, where **x'** is a trial move, and so on, as labeled. This condition for (super) detailed balance as shown in Figure 4.2a was originally stated by Siepmann[98,99]. In his derivation, the degrees of freedom for the initial perturbation were orthogonal to the annealed degrees of freedom, under the adiabatic assumption. Defining the topology of the states is sufficient to derive a slightly more general form, with the assignment of coordinate states added only for clarity. The 3 cases of interest are described in Figure 4.2.

The condition of detailed balance is met only if we choose to enforce the flowrates between states 1 and 3. If all states were accounted for (1,2,3 and 4), then only the condition of balance would be satisfied [9]. In either case, however, the satisfaction of Eq. 129 will ensure a proper distribution of states.

Hierarchical Perturbations

The general scheme is a hierarchical decomposition of move sets. The initial perturbation is designed to be a large move which will cover large regions of configurational space, while the series of trial moves in the 'inner loop' are much smaller, and designed to be an annealing move. The initial trial move, labeled $\boldsymbol{\xi}^{(1,2)}$, is, for all cases here, a vector of uniform variates over some domain $[-\mathbf{d}^{(1,2)}/2, \mathbf{d}^{(1,2)}/2]$, where $\mathbf{d}^{(1,2)}$ is a vector of the same dimension as the complete space (with zero entries for the degrees of freedom that are not sampled). The series of Monte Carlo steps in the inner loop use a different perturbation type, $\boldsymbol{\xi}_n^{(2,3)}$, where *n* is the inner step number, and the domain is $[-\mathbf{d}^{(2,3)}/2, \mathbf{d}^{(2,3)}/2]$.

Figure 4.2 shows schematically the three main ways that the perturbations can differ. For the adiabatic pathway, the initial perturbation is only along one subspace (**x**), while the annealing steps are along **y** coordinate only. For the hybrid pathway, the initial perturbations are in **x**, while the remaining perturbations are in **x** or **y** (or both). Finally, the diagonal path allows for perturbations in both **x** and **y** in both the initial and final step.

The essential feature of the algorithm presented is that the perturbation domains $\mathbf{d}^{(1,2)}$ and $\mathbf{d}^{(2,3)}$ perturbations differ in some hierarchical way. Typically, the initial perturbations will be larger, such that local basins can be traversed. The annealing step uses smaller perturbations to search for nearby low energy states from the initial trial. As long as the total space is covered by the combination of perturbations, meaning that there are no zero values in the vector sum $\mathbf{d}^{(1,2)} + \mathbf{d}^{(2,3)}$, the complete sampling of space is possible.

This approach is not, in and of itself, guaranteed to solve the quasi-ergodic problem, however. The connectivity between states that is generated by perturbation domain $\mathbf{d}^{(1,2)}$ can still limit the accessibility of alternative macrostates. The use of the annealing steps, however, permit larger move sets in the initial trial step that might not otherwise be practical. Table 4.5 describes the types of move sets used for all simulations generated for the present work.

Acceptance Criterion

The forward transition can be defined as a combination of moves described diagrammatically in Figure 4.2. The forward transition probability follows the pathway (1→2→3), which is a combination of the (1→2) transition and the (2→3) transition. While it is possible to require that the reverse transition be along the pathway (3→2→1), the resulting acceptance probability depends on the energy of the initial trial state, which does not improve the acceptance probability. This is shown explicitly in Appendix A. Here, an alternate reverse path is proposed.

To emphasize the difference in forward and reverse paths, we describe this as the POSH pathway (Port Out, Starboard Home). Here, state 4 consists of the reverse trial perturbation, followed by a trajectory which arrives at state 1. This can be described as:

Equation 130

$$p_1\alpha_{12}\alpha_{23}acc_{13} = p_3\alpha_{34}\alpha_{41}acc_{31}$$

where α_{ij} and acc_{ij} are the selection and acceptance probabilities, respectively, of state j from state i. The trial coordinate $\mathbf{q}_2$ is generated as:

Equation 131

$$\mathbf{q}_2 = \mathbf{q}_1 + \boldsymbol{\xi}^{(1,2)}$$

where $\boldsymbol{\xi}^{(1,2)}$ is a uniform deviate vector that perturbs along the (1→2) portion of the pathway. Figure 4.3a shows the forward path construction in detail. The final trial coordinate $\mathbf{q}_3$ is generated using a series of Monte Carlo transitions which will be described shortly, and the reflected trial coordinate $\mathbf{q}_4$ can be defined in terms of the final trial coordinate and the deviate $\boldsymbol{\xi}^{(1,2)}$ used to generate the trial coordinate:

Equation 132

$$\mathbf{q}_4 = \mathbf{q}_3 - \boldsymbol{\xi}^{(1,2)}$$
$$\mathbf{q}_4 = \mathbf{q}_1 + \delta\mathbf{q}_{N_I}$$

where

Equation 133

$$\delta\mathbf{q}_{N_I} = \mathbf{q}_3 - \mathbf{q}_2$$

is the change in position from the trial position to the final trial position after a series of inner steps N_I.

In principle, the coordinate state $\mathbf{q}_4$ need not be defined in terms of the forward random deviate to preserve the property $\alpha_{12} = \alpha_{34}$. The choice of the reflected trial coordinate given by Eqs. 132 and 133 have the convenient property of allowing the transition pathway (4→1) to be defined using the information from the forward trajectory, forming a 'closed loop'. The (2→3) and (4→1) selection probabilities are given by the following transition probabilities:

Equation 134

$$\alpha_{23} = T_{23}^{(N_I)}$$
$$\alpha_{41} = T_{41}^{(N_I)}$$

where $T_{23}^{(N_I)}$ and $T_{41}^{(N_I)}$ are the transition probabilities along the respective Markov chains of length N_I. Solving for the ratio of acceptance probabilities gives:

Equation 135

$$\frac{acc_{13}}{acc_{31}} = \frac{p_3 T_{41}^{(N_I)}}{p_1 T_{23}^{(N_I)}}$$

Eq. 135 obeys detailed balance to the extent that the transition probabilities are correctly estimated. The transition probability of a multistep stochastic walk can be described by the Chapman-Kolmogorov equation:

Equation 136

$$T_{ij}^{(N_I)} = T^{(N_I)}(\mathbf{q}_j \mid \mathbf{q}_i)$$
$$= \int d\mathbf{q}_1 ... d\mathbf{q}_{N_I - 1} \prod_{k=1}^{N_I} t^{(i,j)}(\mathbf{q}_{k-1} \mid \mathbf{q}_k)$$

where *k=0* corresponds to initial state *i* and *k* = N_I corresponds to state *j,* and

$t^{(i,j)}(\mathbf{q}_k \mid \mathbf{q}_{k-1}) = t_{k-1,k}^{(i,j)}$ is the

transition probability from state *k-1* to state *k* (at step *k*). An estimate of this integral over all paths can be made by computing the product of a single series of transitions:

Equation 137

$$T_{ij}^{(N_I)} = \prod_{k=1}^{N_I} t_{k-1,k}^{(i,j)}$$

Eq. 136 is an exact description of the transition probability, which incorporate all possible paths connecting states *i* and *j,* while Eq. 137 is an estimate based on the transition probabilities recorded from a single trajectory.

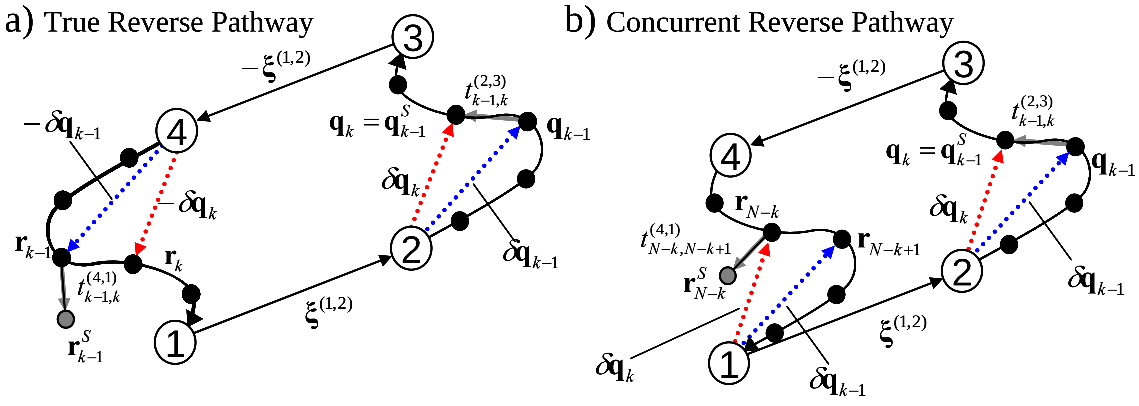


Figure 4.3 *Notation and indexing for transition matrix construction Coordinate states 1,2,3 and 4 are described in the text. The perturbation from 1 to 2 is given by the vector* $\xi^{(1,2)}$*, which also connects states 3 and 4.forward trajectory along the (2→3) pathway is connected by a line with black dots, which represent coordinates along the annealed path. A set of trials at step (k-1→k) is shown for reference. For clarity, the case of an accepted trial move in both forward and reverse transisitions are shown. The white circles are trial moves that are not accepted. For the forward trajectory, the accepted move is a black dot connected with a gray arrow. The gray arrow represents the transition for which a probability is computed. The red and blue arrows show the difference vectors for which the reverse trajectory is constructed. The reverse transition probability connects the k-1 reverse coordinate to an accepted trial coordinate, which is not necessarily the same as the next step in the reverse pathway. The only difference between a) and b) is in the way that the reverse coordinate is constructed using the difference vectors (shown in red and blue) a) True Reverse Pathway b) Concurrent Reverse Pathway.*

Forward Transition Probability

The forward transition probability is estimated using the record of the selection and acceptance probabilities of the single trajectory generated. For the purposes of this work, it is assumed that the selection probability is uniform for all trial moves, so that only acceptance probabilities need to be computed. If a trajectory in the inner loop is generated using the Barker acceptance criterion, the transition probability at the *k*th inner step is:

Equation 138

$$t_{k-1,k}^{(2,3)} = \frac{p(\mathbf{q}_k = \mathbf{q}_{k-1}^S)}{p(\mathbf{q}_{k-1}) + p(\mathbf{q}_{k-1}^T)}$$

where the the trial coordinate $\mathbf{q}_{k-1}^T$ is generated using the uniform deviate:

Equation 139

$$\mathbf{q}_{k-1}^T = \mathbf{q}_{k-1} + \boldsymbol{\xi}_k^{(2,3)}$$

The selected coordinate $\mathbf{q}_{k-1}^S$ is the coordinate resulting from the application of the Barker criterion. A random number is generated over the domain $[0, p(\mathbf{q}_{k-1}) + p(\mathbf{q}_{k-1}^T)]$. If the random number is greater than or equal to $p(\mathbf{q}_{k-1}^T)$, the trial move is accepted, and $\mathbf{q}_{k-1}^S = \mathbf{q}_{k-1}^T$. Otherwise, the trial move is rejected, and $\mathbf{q}_{k-1}^S = \mathbf{q}_{k-1}$.

Rewriting the transitions of Eq. 138 with the Metropolis transition probability gives:

Equation 140

$$\left(t_{k-1,k}^{(2,3)}\right)_{Metropolis} = \delta(\mathbf{q}_{k-1}^S - \mathbf{q}_{k-1}^T) acc_{k-1,k}^{(2,3)} + \delta(\mathbf{q}_{k-1}^S - \mathbf{q}_{k-1})(1 - acc_{k-1,k}^{(2,3)})$$

Dellago uses a similar relation to define the Metropolis action[123]. The usual Metropolis criterion is applied to select or reject the trial coordinate:

Equation 141

$$\left(acc_{k-1,k}^{(2,3)}\right)_{Metropolis} = \min(1, p(\mathbf{q}_{k-1}^T) / p(\mathbf{q}_{k-1}))$$

Notice that the notation and generation of the trial coordinate are identical to that of Eq. 140. In principle, any record of transitions which can be maintained can be used in lieu of either of the two expressions presented. In practice, however, the relations which obey detailed balance have given good results for the model systems studied. Using Metropolis transitions preferable approach for most types of simulations[124], and this is true for the present case as well. For the remainder of the paper, however, the transition elements will be described using the Barker acceptance probability, since the notation is easier to read.

Reverse Transition Estimates

It should be noted that the estimate of the (4→1) transition probability represents a challenging class of problems whereby the endpoints are known, and the calculation of all paths connecting them as described by Eq. 136 needs to be estimated. In general, the estimate of this probability is accomplished through importance sampling. Techniques such as Transition Path Sampling, have gained wide use in generating such estimates [125-127], whereby macrostate endpoints are defined, rather than fixed coordinate states. This is generally a favorable approach, especially since the forward (2→3) trajectory can be thought of as a importance sampled transition path. While Transition Path Sampling is primarily a method for estimating rate constants, it contains similar notions of transitions. Here we present an alternative method for estimating a transition probability, which has been demonstrated to be useful here, but there is clearly an interest in generating more robust and efficient methods for estimating these transition probabilities.

Reverse Transition Pathway (True)

Figure 4.3a describes the reverse transition path construction. Since it most closely mirrors the forward path, we label this as the true reverse transition pathway. To construct the reverse pathway, we first keep track of the displacement from initial trial state at step *k*:

Equation 142

$$\delta\mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_2$$

The reverse path is defined using the displacements from the forward path. The true reverse transition pathway can be estimated using the forward trajectory information in the following way:

Equation 143

$$\mathbf{r}_k = \mathbf{q}_4 - \delta\mathbf{q}_k$$

where $\mathbf{r}_k$ is the *k*th coordinate state in the reverse pathway as constructed in Figure 4.3a. The transition probability from state *k*-1 to *k* is recorded as:

Equation 144

$$t_{k-1,k}^{(4,1)} = \frac{p(\mathbf{r}_{k-1}^{S})}{p(\mathbf{r}_{k-1}) + p(\mathbf{r}_{k-1}^{T})}$$

where $\mathbf{r}_{k-1}^{T} = \mathbf{r}_{k-1} + \xi_k^{(4,1)}$ is constructed using the same perturbation strategy (using domain $\mathbf{d}^{(2,3)}$) as the forward case. The selection of coordinate $\mathbf{r}_{k-1}^{S}$ is follows the same procedure as in the forward pathway. It is important to notice that the selected coordinate $\mathbf{r}_{k-1}^{S} \neq \mathbf{r}_k$, since $p(\mathbf{r}_k)$, can easily become vanishingly small relative to $p(\mathbf{r}_{k-1})$ in the reverse trajectory for complex landscapes, whereas $p(\mathbf{r}_{k-1}^{S})$ is selected according to its probability weight. The coordinate $\mathbf{r}_{k-1} = \mathbf{q}_4 - \delta\mathbf{q}_{k-1}$, forms the anchor point at each step along the reverse pathway, from which a trial coordinate is generated.

One practical consideration when using eqs 18-20 to generate the reverse trajectory is that it requires the storage of the complete forward trajectory $\delta\mathbf{q} = \{\delta\mathbf{q}_1, \delta\mathbf{q}_2, ...\delta\mathbf{q}_{N_I}\}$ prior to generating the reverse trajectory. Since these trajectories are generated using random deviates, this can be accomplished by maintaining a list of the random seeds, rather than an exhaustive storage of coordinate states. Even with this approach, however, it is can be cumbersome to reconstruct the entire reverse trajectory only after the entire forward trajectory has completed.

Concurrent Reverse Transition Pathway

To simplify the storage requirements, an alternative path for the reverse coordinate can be defined

Equation 145

$$\mathbf{r}'_{N_I-k} = \mathbf{q}_1 + \delta\mathbf{q}_k$$

which provides the same connectivity between state 4 and state 1 (See Figure 4.3b). Using this definition of the reverse coordinate path, the following transition can be defined:

Equation 146

$$t'^{(4,1)}_{N-k,N-k+1} = \frac{p(\mathbf{r}'^{S}_{N-k})}{p(\mathbf{r}'_{N-k}) + p(\mathbf{r}'^{T}_{N-k})}$$

This pathway can be generated as the forward trajectory is being generated (since it doesn’t require knowledge of the final trial state). The storage requirements are much less for this pathway, and it is slightly easier to implement. A discussion of the errors introduced by using either of these pathways is in Appendix C.

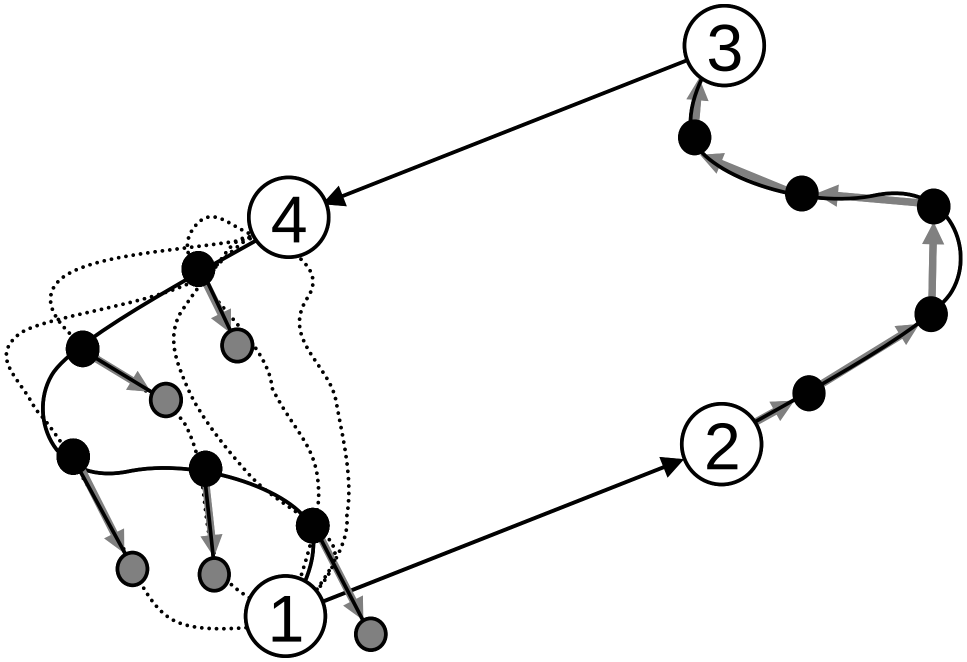


Figure 4.4 Illustration of forward and reverse transition probability calculations. The possible ensemble of paths connecting states 4 and 1 represent putative trajectories that exhibit similar transitions in various stages of the trajectory.

Qualitative Justification for the Reverse Pathway Estimation

The primary motivation for using accepted trial moves (using either method) at each step in the reverse pathway is to maintain numerical stability. Since the reverse pathway is constructed in a region of space that has not been located using importance sampling, as is the case with the forward pathway, reconstructing the path exactly will generate vanishingly small probabilities for even the simplest of landscapes, such as those studied for this work. The fact that the states are no longer connected contiguously may in fact improve the estimate, as a collection of transitions along the reverse pathway is estimating an ensemble of reverse pathways (see Figure 4.4). In fact, the key challenge to improving this sampling strategy is an understanding of how to efficiently and accurately estimate these transition probabilities.

Results and Discussion

Error Metric and Efficiency Considerations

As a general measure of the quality of the sampled distribution versus the true distribution, we can define the following ergodicity metric which is commonly used assessment of sampling quality [104,128,129]:

Equation 147

$$\chi^2 = \frac{1}{N_B} \sum_{i=1}^{\sqrt{N_B}} \sum_{j=1}^{\sqrt{N_B}} \left(G(\mathbf{x}_i, \mathbf{y}_j) - H(\mathbf{x}_i, \mathbf{y}_j) \right)^2$$

where χ^2 is the mean squared error (MSE) over the course of the entire simulation, N_B is the number of bins, $G(\mathbf{x}_i,\mathbf{y}_i)$ is the normalized distribution as described in Appendix B. $H(\mathbf{x}_i,\mathbf{y}_i)$ is the normalized histogram at the square bin centered about $(\mathbf{x}_i,\mathbf{y}_i)$, with dimensions δbxδb.

To compare trajectories, the ratio of MSEs is evaluated using the same number of energy evaluations throughout the entire trajectory, which includes those energy evaluations in the inner loop. For an inner loop of length N_I, the number of energy evaluations required is $2(N_I+1)$ per outer step. We can define a simple improvement metric:

Equation 148

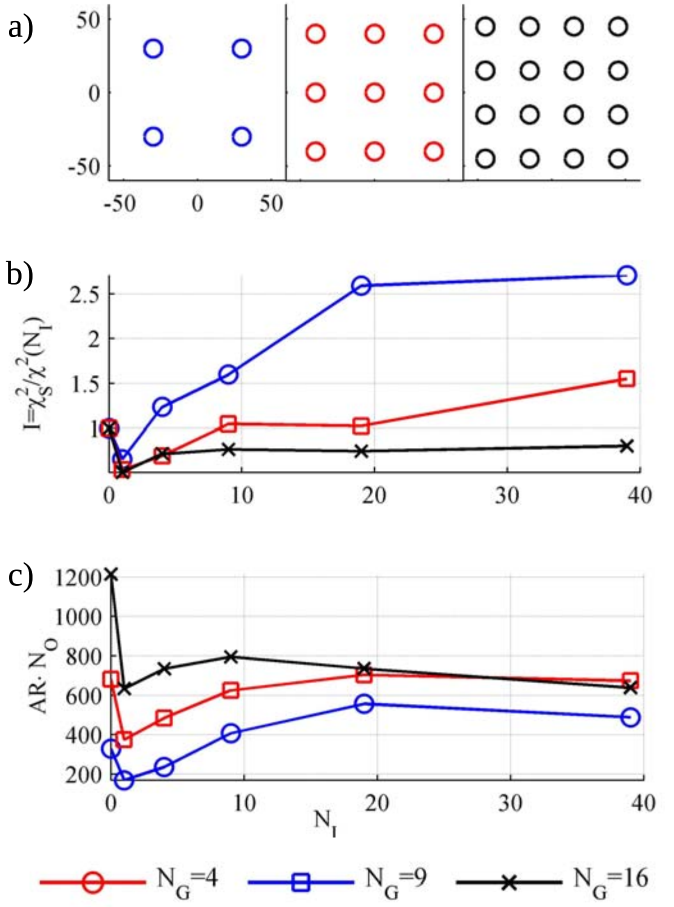
$$I = \frac{\chi_S^2}{\chi^2(N_I)}$$

Where the MSE of a standard simulation, $\chi_S{}^2$ is computed using the number of steps, defined as N_{STD}. In order to ensure a fair comparison, the total number of energy evaluations is held constant, such that $N_O=2N_{STD}$ / (N_I+1), where N_{STD} is the number of energy evaluations of a ‘standard’ simulation (with no annealing step). Thus, while a larger number of inner steps will almost always improve the acceptance ratio, it will not always improve the efficiency of sampling, due to the cost of generating the trial move.

Figure 4.5a shows 3 landscapes, each with evenly distributed arrangements of equivalent basins in the same domain. The basins are identical each with parameters given as landscape G in Table 4.4.

For N_G=4, the improvement decreases to a minimum of 0.8 times that of the standard simulation. For N_I =4, there are simply not enough inner step moves to reliably locate nearby basins because the large initial trial move in the sparse space will very rarely land in a favorable region of space. At N_I = 9, however, the annealing steps begin to locate basins, recovering a nearly equivalent MSE to the standard simulation. At N_I =9-39, a noticeable improvement is observed, which tapers off at N_I =39, since the cost of a trial move is 80 times that of a standard simulation. The total number of accepted outer steps is a good measure of the effectiveness of the sampling, and is shown in Figure 4.8c. The data show that the highest number of outer steps accepted for this landscape is at N_I = 9. The improvement in error at N_I =39 occurs even though there are fewer newly accepted configurations because the trial moves are more decorrelated, which will also serve to improve the statistics.

For N_G = 9, and N_G = 16 a similar drop in efficiency is observed at the lower inner loop settings. The improvement is recovered however, for the N_G = 9 case. For the N_G = 16 case, the standard sampling approach is more effective, as the basins are sufficiently densely packed that a POSH scheme is no longer needed. Note also that the number of newly accepted outer steps is much higher for the standard setting (N_I=0).



N_O/1e3	100	25	10	5	2.5	1.25
N_I	0	1	4	9	19	39

Figure 4.5: Efficiency of POSH sampling depends on the sparsity of minima. a) Schematic of Landscapes sampled Colors correspond to legend. Each circle represents a single gaussian basin (Table 4.1, landscape G) b) MSE versus number of inner steps. c) Number of steps accepted. The number of outer steps in each case is adjusted such that the total number of energy evaluations is the same (100k) for each setting. See table 4.5 for sample settings.

Since the algorithm was motivated by an interest in locating disjoint minima in a sparse space, it is not surprising to see the performance depend strongly on the sparsity of the basins. So, for a given landscape and perturbation protocol, there exists an optimal N_I, which decreases to 1 as the landscapes become less sparse.

These results also suggest that, the initial perturbation should be large relative to the annealing step. The initial perturbation should be large enough cross barriers, while the size of the annealing step is chosen to give good acceptance in sampling the local basin. It is usually straightforward to estimate the size of the annealing step. Often, the length scales emerge quite naturally from knowledge of the system. For example, a typical range for a single dihedral perturbation in a proteins or small molecules is typically less than $2\pi/3$, which is roughly the width of a single χ well.

For the cases studied here, the acceptance ratios for the standard protocols in the first 2 cases (those showing improvement) were less than 1%, which means that the initial perturbation is designed to be ineffective for the standard protocol.

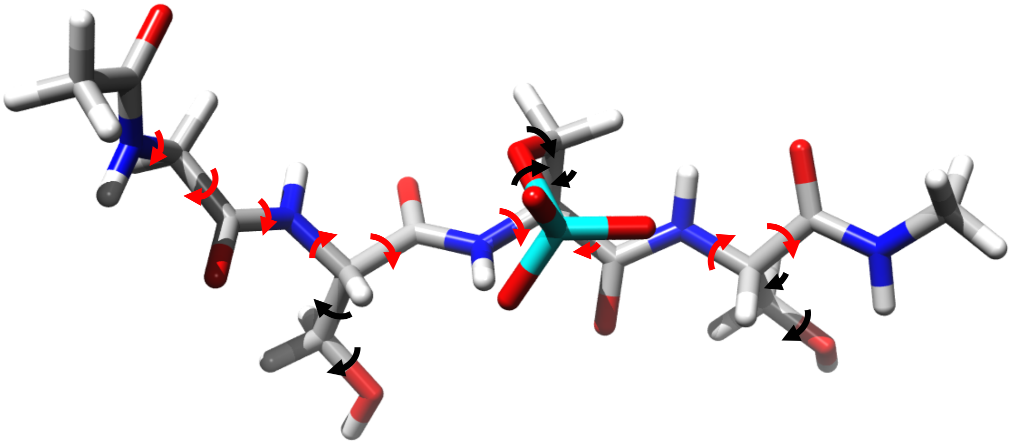


Figure 4.6 – Tetrapeptide model system

Molecular System Application: Phosphopeptide

As a first biomolecular application, we applied POSH to a phosphopeptide: Ace-Gly-Ser-pSer-Ser-Nma, which has been studied previously for forcefield development[130,131]. This system is small but challenging for sampling. The phosphate group forms hydrogen bonds with different combinations of backbone nitrogen groups , and transitions between these conformations are relatively slow in MD simulations.

For this study, only dihedral angles are allowed to fluctuate, excluding the peptide ω angles and capping methyl group torsions. The current implementation is in an SGB[20] implicit solvent model, with an external dielectric of 80 and an internal dielectric of 2. The nonpolar term is from the Levy and Gallicchio[48] model, and the OPLS-AA 2005[49,132]. The phosphate partial charges are nonstandard, and are those used by Groban *et al*[133], based on a study conducted by Wong *et al*[130].

Implementation Details: Modular Perturbations

Perturbations are classified as either backbone moves or sidechain moves. For the current work, a backbone move consists of selecting a single ϕ or ψ angle randomly, and assigning a uniform variate perturbation to that angle over a defined domain. For the initial trial perturbation, the domain is $[-\pi,\pi]$, while the inner step trial moves are over a smaller perturbation domain $[-\pi/10, \pi/10]$.

The sidechain perturbation follows the same prescription regardless of whether it is considered to be an initial trial or an inner step move. It is accomplished by selecting a residue site randomly, and assigning a uniform variate perturbation to each of the χ angles, and polar hydrogens if needed. Here, the each uniform variate is over the domain of either $[-\pi,\pi]$ or $[-\pi/20, \pi/20]$, determined randomly, with the selection probability of either domain given equal probability. This is the same protocol developed in previous work which studies only sidechain fluctuations[121]. The polar hydrogens are sampled over the domain $[-\pi,\pi]$ at every step for which that residue is selected.

At both the initial trial step generation and the inner step moves, either a backbone move or sidechain move is selected with equal probability. This corresponds to the diagonal pathway, since all degrees of freedom are allowed to fluctuate in the (1→2) as well as the (2→3) transitions. To generate the (4→1) transition, the concurrent reverse pathway is used (Eq. 146). At each forward transition, a reverse trial move is generated using the same type of perturbation for the forward perturbation (sidechain or backbone). Both forward and reverse trajectories use Metropolis transitions.

Empirical Corrections: Stationary Transitions

A practical consequence of the reverse pathway estimation in complex systems is that the reverse transitions corresponding to a rejected trial have been observed to result in an error in the estimate the ratio of forward and reverse transitions. In particular, the reverse transition calculations are most susceptible to error, as the construction of the reverse pathway will often pass through sterically hindered portions of configuration space, resulting in anomalously high energies. In these cases, it has proven to be useful to introduce the following empirical correction to the reverse transition probability:

Equation 149

$$t'^{(4,1)}_{N-k,N-k+1} = \delta(\mathbf{q}^{S}_{N-k} - \mathbf{q}^{T}_{N-k+1}) acc^{(4,1)}_{N-k,N-k+1} + \delta(\mathbf{q}^{S}_{N-k} - \mathbf{q}_{N-k}) t^{(2,3)}_{N-k,N-k+1}$$

where $a^{(4,1)}_{N-k,N-k+1}$ is the Metropolis acceptance probability for the reverse transition. This relation simply uses the forward transition probabilities in the reverse trial move when the forward trial is rejected, which prevents the overestimate of the ratio $T^{(N_I)}_{41} / T^{(N_I)}_{23}$, especially in longer inner loop settings. The motivation for the use of this correction comes from the notion of an “ideal transition”. If we consider the ideal forward transition move, it would consist of a series of purely downhill moves. Likewise, the reverse transition pathway would also consist also consist of purely downhill moves. For these 'ideal transitions', the estimate of the ratio $T_{41}^{(N_I)} / T_{23}^{(N_I)}$ would be unity. If the reverse trajectory moves through a very challenging portion of configuration space, it is frequently a better assumption that the ratio of transitions is unity than to compute the ratio of reverse transitions.

MTSMC Acceptance probability

To improve performance, some of the energy parameters $\mathbf{P}_L$ are held at the latent state of the original coordinate, giving the parameter set $\mathbf{P}_L(\mathbf{r}_I)$. Most notably, the long range interactions and Born Radii are held fixed during the inner loop sampling. At the end of each cycle of POSH sampling, the resulting coordinate state is taken to be a trial move and subjected to the Multiple time-step Monte Carlo acceptance criterion:

Equation 150

$$acc_{I,F} = \min\left[1, \frac{p(\mathbf{r}_F \mid \mathbf{P}_L(\mathbf{r}_F))}{p(\mathbf{r}_F \mid \mathbf{P}_L(\mathbf{r}_I))}\right]$$

which has been described previously[121] in more detail. Here we refer to state *I* is the initial coordinate state and state *F* is the final coordinate. The probability $p(\mathbf{r}_i \mid \mathbf{P}_L(\mathbf{r}_j))$ is the Boltzmann factor of the energy evaluated at current coordinate state *i* with latent parameters from coordinate state *j*. For all cases studied here, a single POSH cycle is followed by a parameter update.

To generate comparable trajectories without POSH sampling, a set of standard trajectories was also generated, which maintained the MTSMC sampling. For these trajectories, the same number of inner steps are used in between latent parameter updates. The trial moves for these are those which would be used for the (1→2) moves in the equivalent POSH scheme.

To improve precision and efficiency of sampling, a mixture of POSH and standard sampling was explored. For this scheme, either the standard or the POSH scheme is selected with equal probability, with the number of inner steps set to the same value for either case.

Finally, a set of standard trajectories for which the latent parameters are fully updated at each step are generated. The same prescription is used for the trial moves as in the MTSMC case.

Validation Results

Figure 4.9 shows the distribution of energies from the standard (MTSMC) scheme versus the equivalent POSH sampling scheme. The energy distributions from either scheme are equivalent, which demonstrates that the POSH sampling is able to reproduce the correct distribution in a polypeptide system.

	Standard			***Posh Fraction=0.0***			***Posh Fraction=0.5***		
N_I	β<E>	σ	d/σ_0	β<E>	σ	d/σ_0	β<E>	σ	d/σ
	-149.48	3.71	0.95						
1	-152.95	3.66	0.00	-153.26	3.56	-0.09	-152.96	3.59	0.00
5	-152.97	3.63	-0.01	-152.89	3.73	0.02	-152.86	3.73	0.02
10	-152.49	3.80	0.12	-153.15	3.57	-0.06	-152.85	3.69	0.02
20	-152.59	3.78	0.10	-151.73	3.91	0.33	-152.57	3.79	0.10
50	-152.78	3.71	0.04	-149.42	4.08	0.96	-151.83	4.00	0.30

Table 4.1 - Precision of POSH Sampling for tetrapeptide systems. Simulations of a standard (MTSMC), POSH simulation and mixed POSH/standard are shown. For all simulations b<E> is the average energy for the system, where b=1/kBT, s is the standard deviation of the energy trajectory, and d/s0 is computed as b/s0(<E>- <E>0), where s0 and <E>0 are taken from the Standard simulation where N_I=1 (highlighted in yellow).

Table 4.1 shows that, for the POSH sampling, quantitative agreement can be obtained for settings of up to N_I=10. Nontrivial deviations from the standard distribution are observed at N_I =20, suggesting an upper bound for precision for this particular system. To improve sampling, a 50% mixture of standard trial moves is incorporated. This is intended to improve both precision and efficiency. It is natural to expect an improvement in precision with this approach. The mprovement in efficiency results from the introduction of heterogeneity in the move sets, such that the likelihood of remaining in a kinetically trapped configuration is reduced. The improvement to precision is mixture of standard simulations appears to substantially improve this deviation, but the estimate of the reverse transition probability is clearly limited to a shorter range of steps for the more complex systems. The reasons for this deviation are largely due to the much more rugged landscape involved. It should be noted that the (4→1) transition almost always consists of unphysical configurations, which, for molecular systems can have pathologically large energies, to the point that even roundoff error can become a factor in the reverse path estimate. The introduction of the empirical correction to the transition estimate appears to substantially improve the quality of the sampling, largely due to the effect described above.

Figure 4.7 shows the rates of convergence to the final energies. To make the trajectories comparable, the number of energy evaluations (excluding the latent parameter updates) are plotted along the x axis. For a standard simulation, there are $N_E = N_O$ evaluations, and for a POSH simulation, there are $N_E = 2N_O\,(2N_I+1)$ energy evaluations. Since a single posh cycle is run prior to updating the Born radii according to Eq. 150, the ‘standard’ trajectory is that which uses a single inner loop step (with no posh sampling) before updating the Born radii.

The rate of convergence is fastest for the mixture of POSH and standard simulations. For the low inner steps settings, the mixed simulation appears to be converge roughly 2 times faster than a standard simulation. The POSH sampling without the mixture does not display improvement for *N*$_I$=1, but does show improvement for *N*$_I$=5. At *N*$_I$=10, all methods perform roughly equivalently, and both POSH protocols fail to show improvement at *N*$_I$=10, which is also where the precision begins to break down.

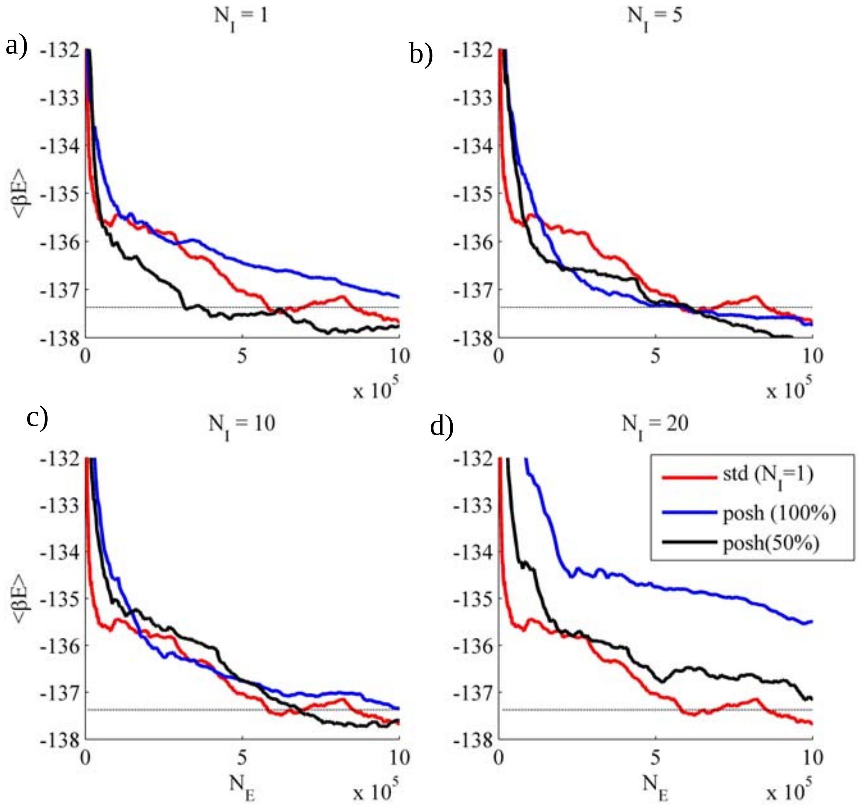


Figure 4.7 – Rates of Convergence. Block Averages are computed and plotted against the number of energy evaluations per datapoint.

Since this system was previously well sampled with standard MD approaches, substantial improvements in efficiency were not expected, since it was selected to evaluate the sampling precision, so it was surprising to observe any improvements at all. It was equally surprising was to see that, for cases where POSH did not show improvement, the mixture proved to be faster than either the standard or POSH sampling alone, suggesting that the synergy of the combined sampling is a useful design feature.

	Experiment	Wong *et al* Simulation	Standard (MTSMC)	Standard (no MTSMC)	
	<J>	*<J>*	*<J>*	*<J>*	
Ser1	5.63(0.12)	4.12(0.03)	3.24(0.04)	3.36(0.02)	
Ser2	6.65(0.12)	6.62(0.11)	7.67(0.02)	7.25(0.01)	
pSer3	5.48(0.12)	5.73(0.10)	5.52(0.05)	5.86(0.02)	
Ser4	6.93(0.12)	7.37(0.03)	7.46(0.01)	7.46(0.01)	
POSH Sampling					
	NI=1	*NI=5*	*NI=10*	*NI=20*	*NI=50*
	<J>	*<J>*	*<J>*	*<J>*	*<J>*
Gly1	3.23(0.11)	3.22(0.10)	3.16(0.14)	3.12(0.08)	3.21(0.06)
Ser2	7.69(0.04)	7.66(0.08)	7.62(0.06)	7.66(0.09)	7.60(0.06)
pSer3	5.59(0.17)	5.58(0.17)	5.45(0.10)	5.51(0.04)	5.61(0.11)
Ser4	7.45(0.03)	7.50(0.02)	7.45(0.02)	7.46(0.02)	7.41(0.02)

Table 4.2 - J coupling Data

Physical Observables

J couplings were computed using the Karplus equation:

Equation 151

$$\left\langle {}^{3}J \right\rangle = \frac{1}{L}\sum_{i=1}^{L}\left(A\cos^{2}\theta_{i} + B\cos\theta_{i} + C\right)$$

where <*J*> is the average NMR J coupling value, *L* is the length of the simulation, and θ_i is the dihedral angle of the *i*th snapshot, of the H-N-C_α-H bond. It is compared to experiments and simulations as reported by Wong and Jacobson[130]. Of all of the simulations, we expect the standard (MTSMC) values to be most reliable, since it contains data from all inner steps settings. We find good agreement between the standard simulation (with MTSMC) and experiment, most notably in the pSer coupling, which is most sensitive to the phosphate interactions. We obtain poor agreement with experiment for the Gly coupling, but this anomaly is also observed with the simulation data, and is due to the fact that *J* couplings are not well defined with regard to Glycines.

	Wong *et al* Simulation	**Standard (MTSMC)**	**Standard (no MTSMC)**		
	% HB	**% HB**	**% HB**		
Gly1	56(04)	63(03)	04(00)		
Ser2	51(04)	62(03)	05(01)		
pSer3	60(04)	80(04)	14(01)		
Ser4	08(02)	09(03)	09(01)		
POSH Sampling					
	NI=1	*NI=5*	*NI=10*	*NI=20*	*NI=50*
	% HB	**% HB**	**% HB**	**% HB**	**% HB**
Gly1	61(13)	54(11)	59(06)	54(07)	51(05)
Ser2	59(12)	53(10)	59(06)	54(06)	50(06)
pSer3	73(14)	67(13)	75(08)	69(08)	66(08)
Ser4	17(11)	20(11)	11(04)	15(04)	08(02)

Table 4.3 - Fraction of Phosphate hydrogen bonded to amide hydrogen by residue

The agreement with the molecular dynamics simulation is good, considering that a different forcefield and implicit solvation model were used. We obtain good agreement of the *J* couplings with the standard (MTSMC) simulation as a control, and notice also that the observables appear to me reasonable across a broader range than that observed with the control. This is because the control is, in general, a stricter measure of sampling precision than the experimental measures. The experimental measures, however, help to validate the overall protocol, which include the forcefield and implicit solvent used.

The hydrogen bonding fractions between the phosphate groups and the amide groups are in table 4.3.

Conclusions and Future Directions

We have presented a sampling protocol that allows for efficient sampling of sparsely distributed basins, such as those that are encountered in complex biomolecular energy landscapes. The protocol obeys detailed balance to the extent that the transition probabilities are correctly estimated. Three variants of the sampling protocols were presented in terms of pathways: 1) adiabatic, 2) hybrid, and 3) diagonal. Each pathway provides adequate performance, and have useful practical motivations, but the diagonal and hybrid approaches are more robust for longer inner loop protocols. Two reverse pathway constructions were presented, and appear to be equivalent in terms of precision, with a slight preference given to the concurrent reverse pathway due to the ease of implementation. The algorithm performs most efficiently on very sparse energy landscapes. It has been implemented in a realistic biomolecular system, and a range of precision and efficiency has been established. A two-fold efficiency is relatively straightforward to obtain, in both model systems and the physical system studied.

For future work, implementation and performance in complex systems, particularly biomolecular, will be helpful to determine how well the algorithm performs. Since the algorithm obeys detailed balance, it can be combined with other methods, such as hybrid Monte Carlo, and Multiple Time Step Monte Carlo to optimize the efficiency of sampling of a complex system overall. Improving the transition probability estimates is an ongoing effort, and future work will address some of the current limitations of the approach. A better estimate could remove systematic biases in the more approximate schemes, as well as limit the number of trial moves used to generate the reverse transition estimate. Finally, an extension of this algorithm to parallel systems could provide better exchange rates between replicas.

Appendix A: Alternative Acceptance Probability

An alternative to the POSH pathway is presented to illustrate the limitations of the standard acceptance criterion. Consider the reverse pathway which passes through the same coordinates as the forward trajectory. The condition for detailed balance is then:

Equation 152

$$p_1\alpha_{12}\alpha_{23}acc_{13} = p_3\alpha_{12}\alpha_{32}acc_{31}$$

The selection probabilities along the (2→3) and (3→2) are given by the transition probabilities:

Equation 153

$$\begin{aligned}\alpha_{23} &= T_{23}^{(N_I)}\\ \alpha_{32} &= T_{32}^{(N_I)}\end{aligned}$$

If each inner step obeys detailed balance the the resulting detailed balance condition is satisfied:

Equation 154

$$p_2T_{23}^{(N_I)} = p_3T_{32}^{(N_I)}$$

Combining Eqs. 152-4 gives:

Equation 155

$$\frac{acc_{13}}{acc_{31}} = \frac{p_3T_{32}^{(N_I)}}{p_1T_{23}^{(N_I)}} = \frac{p_2}{p_1}$$

This criterion is, in general inefficient, since it depends on the energies of the original and the initial trial.

Appendix B: Model System

Consider a 2 dimensional Gaussian distribution:

Equation 156

$$g_i(x,y) = w_i \exp\left(-u_i(x,y)\right)$$

where x and y are scalar variables, and the index i refers to the ith parameter set defining the distribution. Models of this sort have been used to analyze biomolecular systems, due to the simple evaluation of the configuration integral and other observables[134,135]. The potential is simply a harmonic potential:

Equation 157

$$u_i(x,y) = \gamma_i \left[x-(x_0)_i \quad y-(y_0)_i\right] \mathbf{R}_{\theta_i}^T \begin{bmatrix} (\sigma_1)_i^{-2} & 0 \\ 0 & (\sigma_2)_i^{-2} \end{bmatrix} \mathbf{R}_{\theta_i} \begin{bmatrix} x-(x_0)_i \\ y-(y_0)_i \end{bmatrix}$$

where γ is a force constant, x_0 and y_0 are the coordinates of the minimum, and the σ_1 and σ_2 are the standard deviations along the principal axes of the distribution. $\mathbf{R}_\theta$ is a matrix which rotates the principal axes of the distribution by an angle θ:

Equation 158

$$\mathbf{R}_{\theta_i} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$$

The entire set of parameters defining a single Gaussian term, then, are given as:

Equation 159

$$\mathbf{p} = \begin{pmatrix} x_0 & y_0 & \sigma_1 & \sigma_2 & \theta & w & \gamma \end{pmatrix}$$

One advantage to using potentials of this type is that the free energy of each basin can be computed analytically. Also, potentials of mean force can be analytically computed.

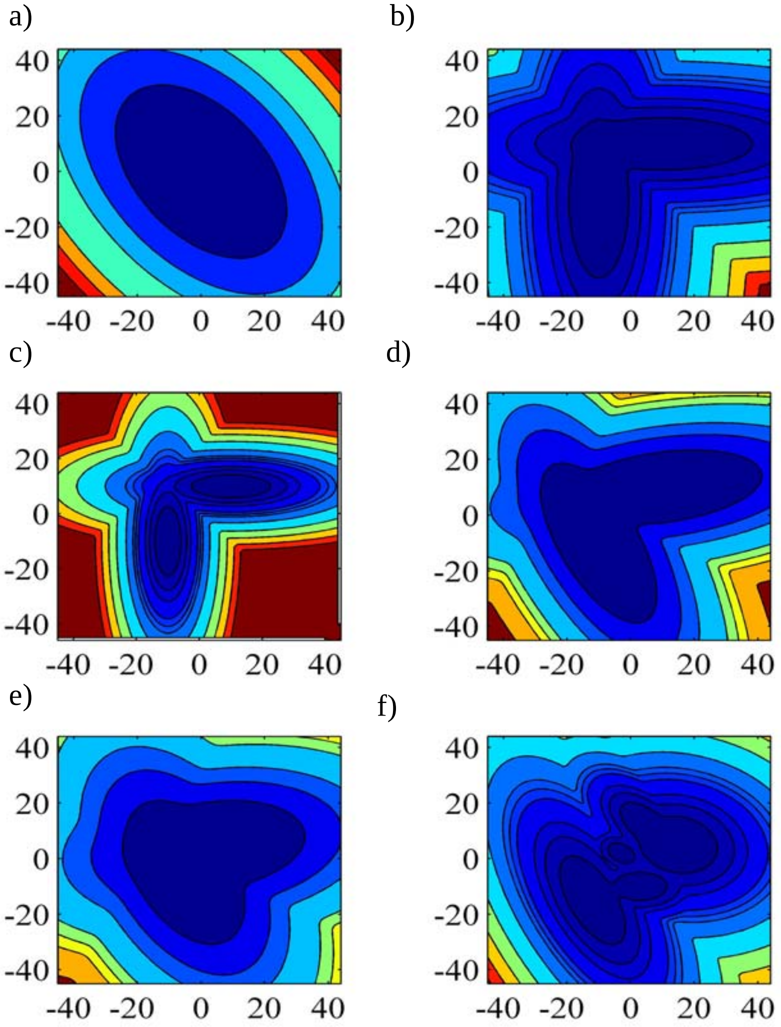


Figure 4.8 - Landscapes Studied. 2D contours of landscapes as described in text with parameters given in Table 4.5. For all plots, x is along horizontal axis. a) Single Basin b) Orthogonal Disjoint Basins c) Orthogonal Accessibe Basins d) Acute Basins e) Acute Basins f) Multiple Basins.

Potentials can be superimposed to generate nonlinear behavior:

Equation 160

$$G(x,y\,|\,\{\mathbf{p}\}_{N_F}) = \sum_{i=1}^{N_F} g_i(x,y)$$
$$U(x,y\,|\,\{\mathbf{p}\}_{N_F}) = -\ln(G(x,y\,|\,\{\mathbf{p}\}_{N_F}))$$

For all cases presented, the temperature is unity.

Landscape #	x_0	y_0	σ_1	σ_2	w	θ/π	γ
A	0	0	3	5	1	0.2	0.2
B	-10	-10	4	12	1	0	9
	10	10	12	4	1	0	9
C	-10	-10	4	12	1	0	1.8
	10	10	12	4	1	0	1.8
D	-10	10	4	10	1	0.15	1
	10	10	4	10	1	0.55	1
E	-5	5	5	9	1	0.15	1
	5	5	5	9	1	0.55	1
F	-12	-15	6	14	3	0.15	6
	5	12	5	10	4	0.25	9
	3	-10	5	3	2	0	3
	15	5	10	8	3	-0.1	6
	-3	2	2	3	1	0.3	3
G	XX	XX	3	3	1	0.0	9

Table 4.4 – Parameters of Landscapes as described in Appendix B.

Appendix C: Model System Studies

Since the algorithm contains the possibility of sampling two subspaces hierarchically, a two dimensional surface becomes a convenient model system for studying the precision of sampling (see Figure 4.5). These model systems are superpositions of 2D Gaussian distributions, whose partition functions are known analytically. Appendix B describes these in detail, and Table 4.4 lists the parameters used. These landscapes were designed to have features that are thought to influence the sampling. One of the features is the symmetry/asymmetry of the surface, which can play a role in due to the symmetry in the reverse path construction. Another feature of interest is the distribution of local minima, either as quasi-ergodic or accessible basins. Landscapes with anharmonic saddle points and complex features were also included to mimic rough landscapes in a way that is intended to make errors in sampling more easily detected.

Figure	B	$\mathbf{d}_x^{(2,3)}$	$\mathbf{d}_y^{(2,3)}$	$\mathbf{d}_x^{(4,1)}$	$\mathbf{d}_x^{(4,1)}$	db	Path Type	Reverse Pathway
8	45	22.5	0	0	9	0.9	Adiabatic	True
8	45	22.5	0	9	9	0.9	Hybrid	True
8	45	22.5	22.5	9	9	0.9	Diagonal	True
9	45	22.5	22.5	9	9	0.9	Diagonal	Both
	60	60	60	12	12	0.25	Diagonal	True

Table 4.5 – Simulation settings for simulations in figures. Parameters of Landscapes as described in Appendix A. *B* represents the lower and upper bound of the landscape in both the domains of both x and y. The values X and X are the lower and upper bounds of the initial (1→2) perturbation for x and y, respectively, and X and X are the lower and upper bounds for the (2→3) perturbation for *x* and *y*, respectively. *db* is the square bin width of the 2D histograms collected.

Choice of Perturbation Scheme

In applying the POSH sampling to a system of interest, many considerations drive the choice of subspace partitions, but the choice will be limited to the types listed in Figure 4.2. It is worthwhile to notice the design considerations involved in each of the perturbation types.

The adiabatic pathway is convenient in cases where the subspaces are thought to be loosely coupled. The adiabatic pathway is also of importance because it is the fundamental idea from which more nuanced descriptions emerge. In some cases, there can be a significant computational advantage to sampling along one subspace while holding the remaining degrees constant. This is certainly true of backbone and sidechain sampling, where, for example, a trial backbone coordinate can be screened for sidechain steric clashes outright before generating a costly Markov chain. The adiabatic pathway is useful when the potential can be evaluated in the annealed subspace, (**y** as shown in Figure 4.2a) much more rapidly than in the perturbed coordinate (**x** as shown in Figure 4.2a). These considerations are not feasible using either of the two other strategies.

The hybrid pathway is named such because it can be used (for example) in concert with standard hybrid Monte Carlo scheme. Here, the choice of perturbation coordinates can be a subspace of 'interesting' coordinates, while the set of annealed steps can be along all degrees of freedom (such as would be the case if a dynamical propagator was used). In the case of biomolecules, for example, the torsions could be perturbed in the initial step, with each step in the inner loop generated according to an HMC scheme (allowing all bonds, angles and torsions to fluctuate).

Finally, the most general of the pathways is the diagonal pathway. For this pathway, all degrees of freedom are perturbed in both steps, with the only difference being in the magnitude of perturbations.

For Figure 4.6, the MSE of the adiabatic pathway with $N_I = 1$ is defined as $\chi_A{}^2$ and calculated for each of the pathways. The MSE, $\chi^2(N_I)$ for each value of N_I is computed and the normalized value $\chi^2(N_I)/\chi_A{}^2$ is a measure of the error relative to a standard error. Table4.5 details the simulation conditions of the landscapes studied. As expected, the diagonal sampling scheme emerges as the most accurate of approaches across all landscapes shown here.

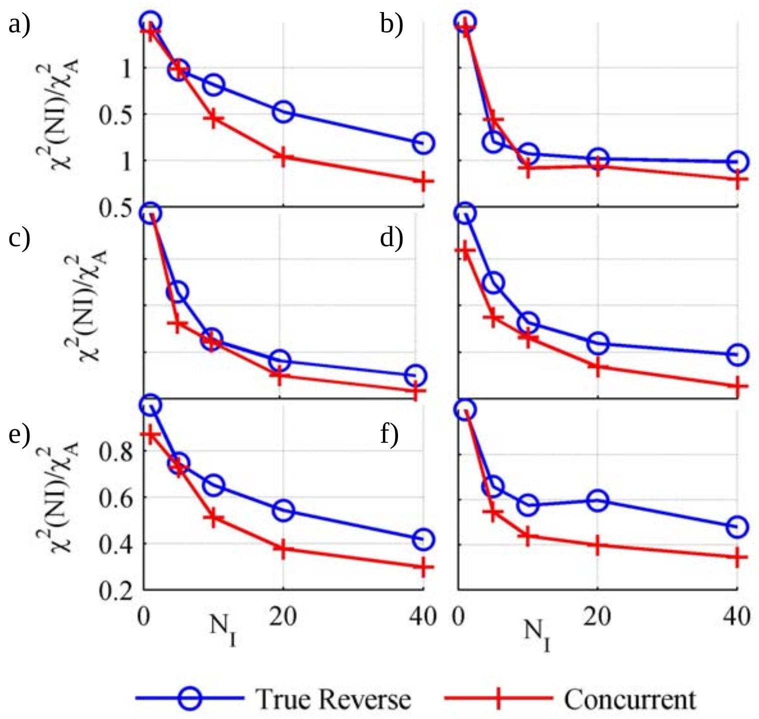


Figure 4.9: Accuracy of true and concurrent reverse paths are nearly equivalent. Each letter corresponds to the landscapes shown in Figure 4.5. MSE errors are normaliszed to the True Reverse pathway with NI=1 for each landscape. Simulations were run with *NO*=40k steps. See Tables 4.4 and 4.5 for additional settings.

The hybrid approach also performs extremely well, suggesting that allowing all degrees of freedom to fluctuate in the annealing steps is a key to improving precision. The hybrid pathway often outperforms the diagonal pathway for smaller inner step settings, but this may be due to the fact that the initial perturbation of the hybrid case is in fact much smaller, since it only spans 1 dimension. At larger values of N_I, the hybrid and diagonal pathways perform nearly identically. While the adiabatic pathway performs very well for smaller values of N_I , it exhibits systematic bias as the number of inner steps is increased, due to the constrained sampling of the annealed coordinate. For shorter trajectories (N_I < 10 for the cases here), the difference in error is negligible for all pathways. Any of these approaches, for relatively short trajectories, could be applied to more complicated systems with confidence, but in cases where longer inner loop trajectories are needed, the hybrid and diagonal pathways are more robust.

Choice of Reverse Pathway Construction

The estimate of the transition probability using the forward trajectory information is a common and relatively well understood method for estimating the forward transition probability[5]. Since the reverse pathway construction is the novel feature of this approach, it is worthwhile to compare the reverse pathway construction methods. Figure 4.7 shows the MSE between the landscapes of Figure 4.5 and the sampled trajectory. For Figure 4.7, the MSE of the true reverse pathway with N_I = 1 is defined as ${\chi_T}^2$ and a ratio $\chi^2(N_I)/{\chi_T}^2$ is calculated for each of the pathways. For all cases, the relative MSE steadily decreases with the number of inner steps. The concurrent reverse pathway appears to generate slightly smaller MSEs. The performance across this range of landscapes suggests that either choice would be sufficient. This is somewhat surprising, since the shape of the reverse pathway is completely different for each case, and it is tempting to think that the shape of the landscape could have a more profound effect on the error. This does not appear to be the case, however, in these test cases. Since the concurrent reverse pathway is both reliable and easier to implement, it is likely to be used more widely.

Chapter 5

Applications, Design Philosophy and Future Directions

One goal of this work is to provide a highly efficient tool for sampling configuration space of proteins in a variety of contexts. It has been part of the philosophy to introduce efficient Monte Carlo algorithms, establish that the algorithms are efficient, precise, and robust, and provide access to daring users of newly developed computational biophysics software. I would like to lay out the overall design philosophy, and explain the where new features are most easily implemented. I would also like to take this opportunity to point to future design features that could be implemented with relative ease.

Design Philosophy

Current Design

The sampling framework is designed to incorporate a mixture of POSH and non POSH trial moves. The motivation for this is several fold. The primary motivation is that the success of a Monte Carlo strategy is largely dependent on the quality of the geometric perturbations, and the inclusion of POSH sampling should provide enhancements. For this reason, a standard scheme should be accessible, which is guaranteed to generate high precision statistics, and be robust (if not optimal) for all parameter settings. Of course, it is hoped that the use of POSH move sets will provide enhanced sampling, and a simple setting is made available to adjust the fraction of times a POSH trial move is generated versus a standard trial move. For even simple biomolecular systems, it should be noted that a 50% mixture of standard and POSH moves provides improvements over either settings.

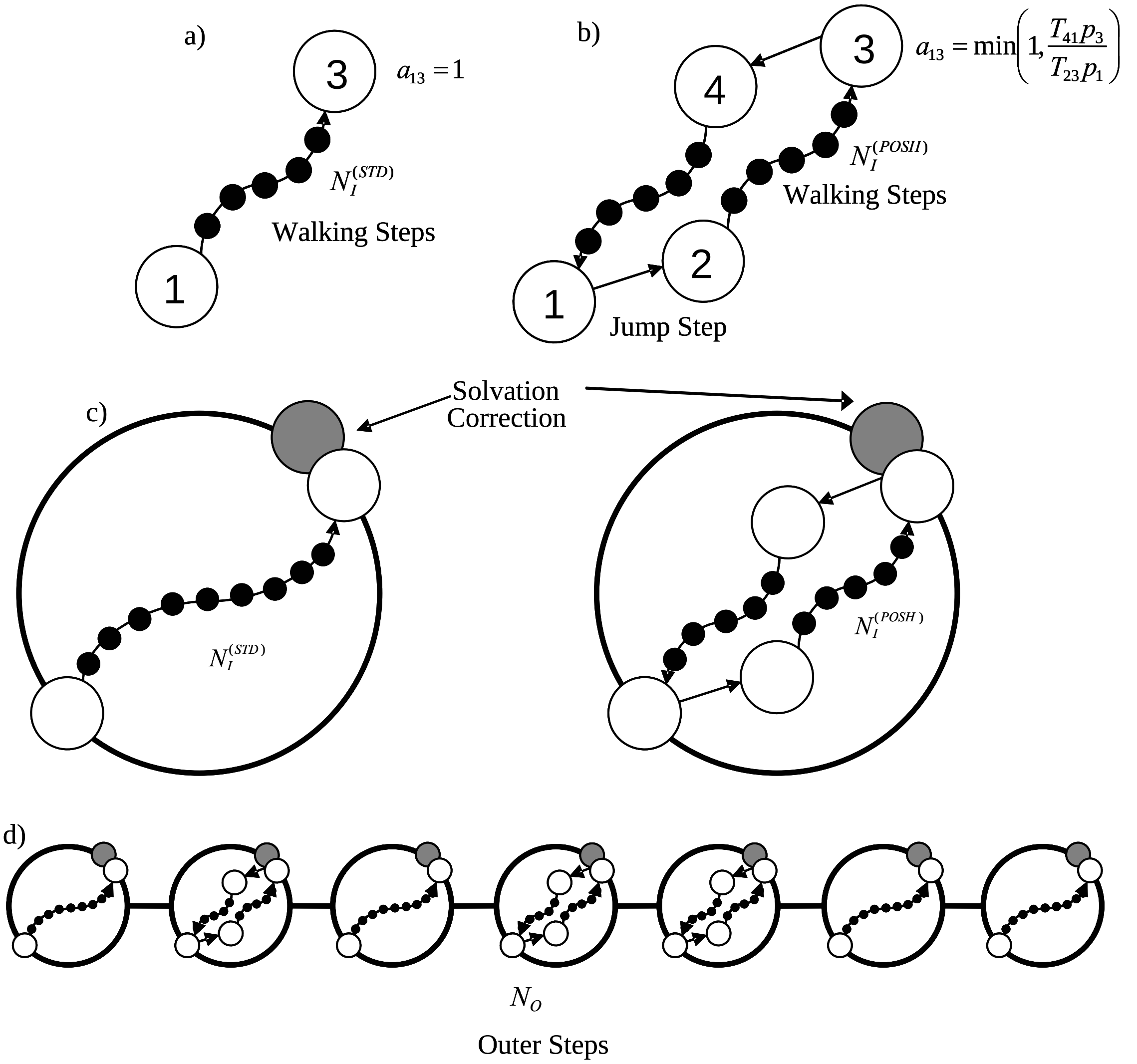


Figure 5.1 – Schematic of sampling loops. a) a standard simulation of length $N_I^{(STD)}$ is shown as a special case of POSH sampling with the initial perturbation set to zero and the final state accepted with probability 1 b) The POSH sampling scheme, with $N_I^{(POSH)}$ walking steps. c) An example of a chain of states using either POSH or STD schemes within the solvation framework.

Figure 5.1 shows a simple analogy between a Markov Chain generated with a standard procedure versus a POSH chain. If we consider the (1→2) trial move as a 'jump' move, with the following steps as 'walk' moves, then a standard trajectory is can be viewed as a POSH cycle with the (1→2) move set to be zero. This modifies the acceptance probability to 1 also.

The *jump* and *walk* moves are contained within the same subroutine, and is currently designed to be a menu of geometric perturbations. The idea is to be able not only to call different jump and walk settings from the configuration file, but to easily modify the code as new perturbation types become available.

The Solvation Envelope

The solvation optimization of the sampling remains as the primary source of the optimization. From the simple studies in Chapter 2, a 10 fold improvement was easy to observe, and it is likely that the improvements can be even better with careful parameter tuning. The current design is to nest the core sampling approaches (either POSH or standard) within the solvation bookkeeping machinery such that this optimization is automatically taken care of when introducing new *jump/walk* options. Figure 5.2 illustrates how the settings are currently used.

Extensions

Small Molecule Sampling: A relatively new development from Ken Borrelli and Victor Guallar is the incorproration of small molecule sampling in the context of the sampling framework described.

Rigid Body/Domain Sampling: The small molecule module contains functionality for sampling rigid body displacements between chains. This could conceivably be extended to the displacements of secondary structures as well.

Multiple Loop Sampling: The incorporation of multiple loops in the sampling protocol requires no new algorithmic work, either with regard to loop closure or Monte Carlo sampling, and should work well once implemented.

Protonation State Sampling: It is also hoped that protonation states can be sampled in the near future, providing an efficient constant pH simulator.

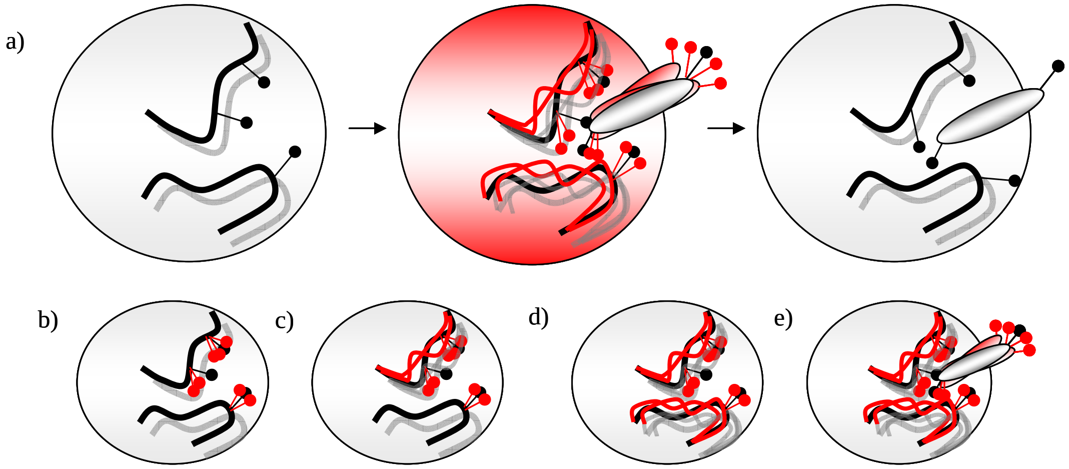


Figure 5.2 – An *apo* simulation is run under a variety of constraints to explore configurations that may lead to *holo* configurations. A) leftmost structure is native *apo* state, middle structure is an arbitrary simulation protocol, leading to a hypothesis about a binding mode of a *holo* structure, shown on the right. Types of simulations are shown in b-e b) sidechain trajectory, as described in Chapter 2, c) single loop and adjacent sidechains (current functionality) d) multiple loops and adjacent sidechains (future implementation) e) multiple loops, sidechains, and ligand. Ligand sampling functionality is available, courtesy of K. Borrelli *et al*

Steric Screening Options

Steric screening is a key element in efficient Monte Carlo routines, and functionality exists currently accomplish steric screening, as described in the sidechain sampling routines described in Chapter 2. Due to the design of the POSH sampler, however, steric screening of the trial state can occlude high energy trial states. The steric screening functionality was therefore disabled in order to establish the validity of the POSH sampler. As optimal sampling ranges are identified, it should also be a priority to reincorporate steric screening in some coherent way.

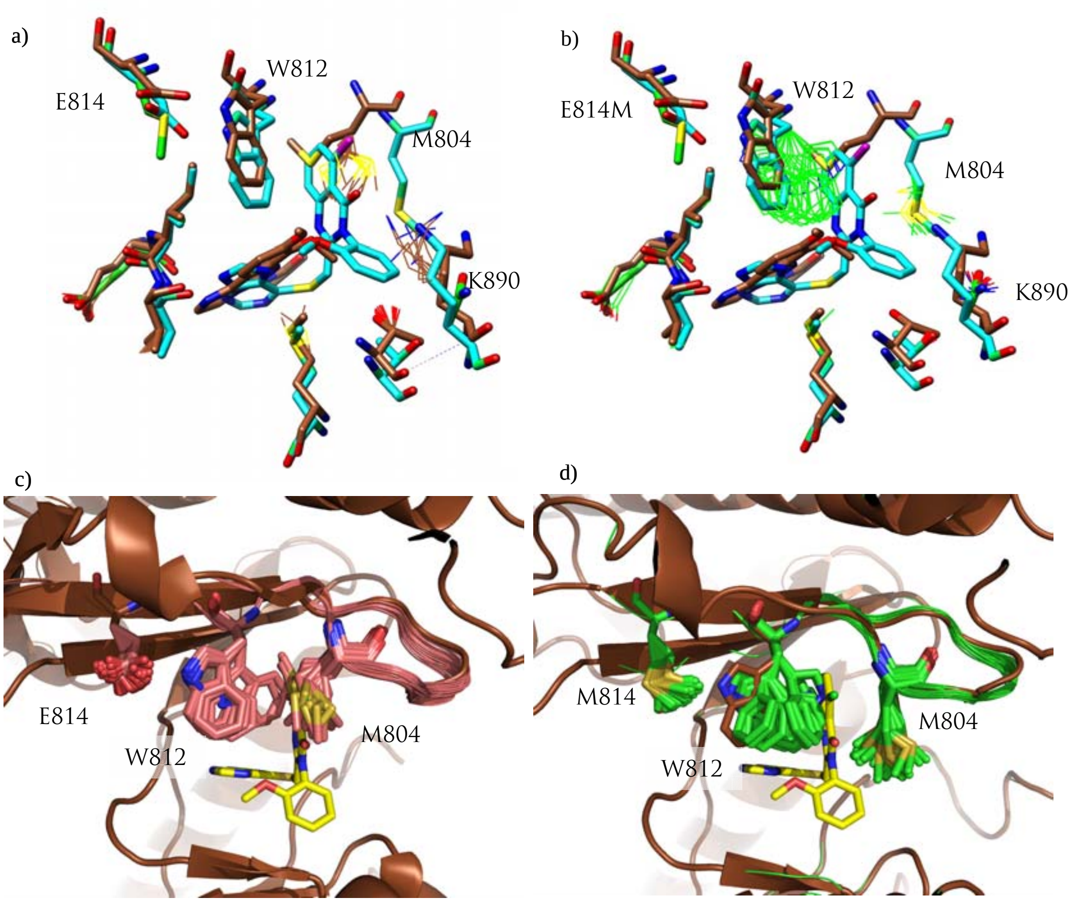


Figure 5.3 – PI3 Kinase Preliminary Studies a) original sidechain studies of γ isoform predicted that M804 played a role in the dynamics of the binding pocket. B) simulation of a δ isoform with point mutations throughout the protein, but not in the first shell of the protein binding pocket. The dynamics of the Trp812 are altered by the mutation from Glu814Met in the second shell of the binding pocket. C) Incorporation of backbone dynamics of single loop in g isoform provides further detail into the nature of the flutuations between Met804 and Trp812, residues which are known to play a role in determining specificity d) simulation of δ isoform also reveals significant fluctuations in Trp812, which leads to enhanced binding affinity. Current efforts to quantify the dynamics of this system are underway.

Applications

Constrained Sampling and Conformational Selection

The idea behind many of the sampling approaches is to take a small functional portion of the protein, and simulate this portion exhaustively such that alternative conformations may be observed. In most cases, this means sampling this portion of the protein at a higher temperature. Due to the nature of the constraints imposed, this is often possible without disrupting the structure to wildly unphysical configurations, as would be the case if the full protein were simulated. The constraints we impose are severalfold, depending on the type of simulation being run. Figure 5.2 illustrates the types of simulations that might be run to explore alternative configurations. All settings currently restrict the ω angles, bond angles, and bond lengths to their native state. Many of these constraints may introduce artifacts which may bias the sampling towards near native configurations, which is one motivation for running higher temperature simulations. As was the case when generating the sidechain trajectories described in Chapter 2, it is noted that even high energy configurations can often be of interest, as these high energy states may encounter stabilization upon the presence of a ligand.

PI3 Kinase

The exploration of PI3 kinase functionality is an ongoing effort in collaboration with the Kevan Shokat group at UCSF[69]. While much of the data is quite nascent, it has become an interesting system for our functionality. The PI3 kinase family is a set of targets implicated in a wide range of diseases, including inflammatory conditions, thrombosis, and cancer. There is a keen interest in the Shokat group to elucidate the structural basis of this functionality. It has also been an interesting companion system to study as new functionalities emerge, and allow for more detailed study of the system. The original apo binding pocket that was studied using the sidechain sampling of Chapter 2 is in Figure 5.3a. An interesting phenomenon that was noticed was that the sidechain trajectory of W812 changed significantly upon point mutations which were more than one solvation shell away from the binding pocket. The simple sidechain trajectory

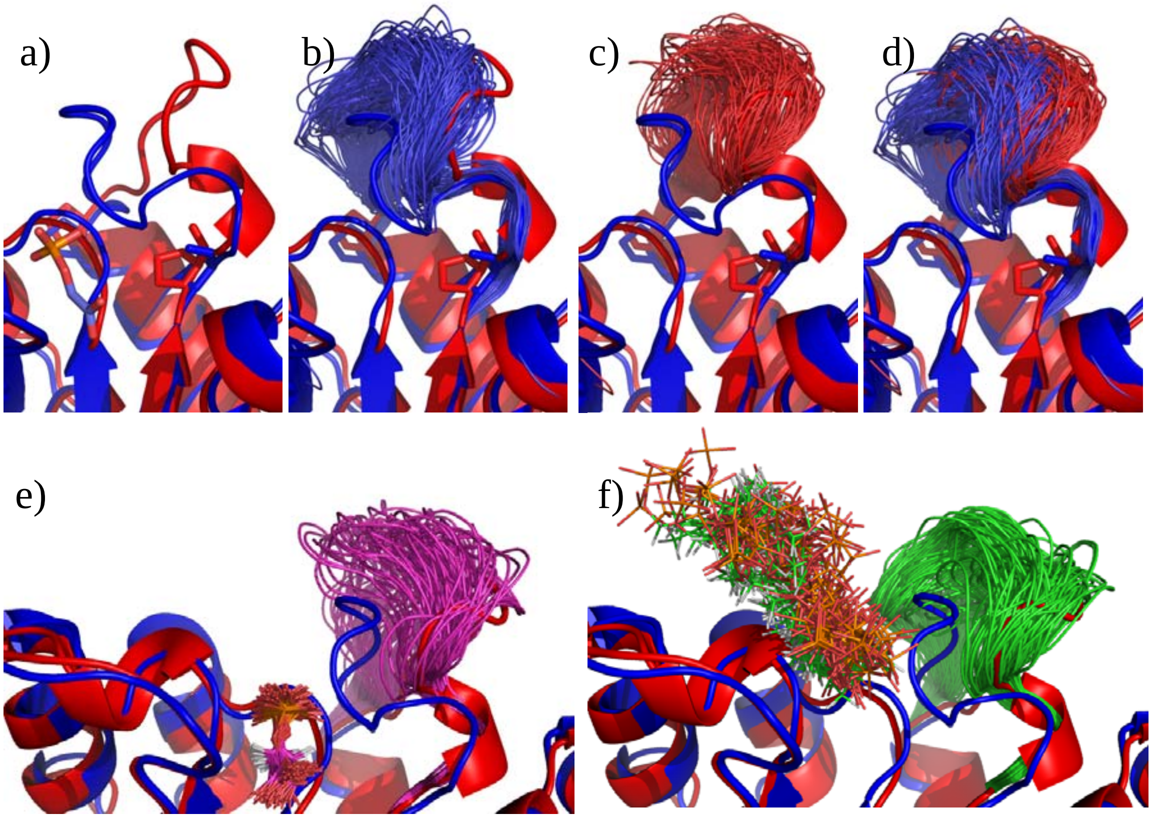


Figure 5.4 –Ensembles of TIM loop a) open *apo* configuration shown in red, and closed *holo* configuration is shown in blue, bound to PGA b) *apo* closed loop ensemble c) *apo* open loop ensemble d) open and closed ensembles appear to have overlapping regimes e) PGA placed in the *apo* open configuration and simulated. Only internal degrees of mothion are sampled. E) Same simulation conditions, but with translational and rotational degrees of freedom allowed to fluctuate.

provided the hypothesis. Once the loop and sidechain functionality became available, we were able to generate *apo* binding pocket trajectories with the β hairpin allowed to fluctuate. Current efforts are underway to quantify the fluctuations within the selectivity pocket (the region between W812 and M804), and to correlate this with known binding affinities as provided by the Shokat Group.

The Canonical Case: Triose Phosphate Isomerase

The dynamics of the catalytic loop of Triose Phosphate Isomerase form are well studied, and have proven to be a good model system for understanding the algorithms.[136] The mechanism is well understood as a latching mechanism. The data presented here is a qualitative demonstration that the method has promise, and current efforts to further quantify these dynamics are underway. Figure 5.4a-d shows these simulations, run at high temperature, and sufficient overlap of configurations suggests that the conformational selection hypothesis may have some applicability in the study of these systems.

An additional demonstration of functionality is show in Figures 5.4e and 5.4f. The open configuration is simulated with the PGA ligand placed in the native holo configuration and simulated. While this data is also preliminary, it is provided to show that the ligand sampling functionality provided by Ken Borrelli is working correctly, and that a much richer variety of systems are also accessible through this methodology.

Future Directions

A Key Advance in the Making: POSH and Hybrid Monte Carlo

The POSH sampling approach presented in Chapter 4 lays a theoretical foundation for a new way of designing Monte Carlo move sets. It is the hope of the author that the developments of Chapter 4 represent a strong first generation of sampling approaches, and that the methodology be allowed to grow into new, more efficient methods in future work. There are two limitations to the initial work presented. The first is that the construction of the reverse pathway will limit the efficiency by at least a factor of 2, which narrows the parameter space for observed improvements. The second limitation is in the existence of an upper bound for the number of inner steps. We were able to observe marked improvements in efficiency and almost complete insensitivity to the number of inner steps for the simple model systems. In the real systems however, the robustness and efficiency was found to be somewhat limited, due mostly in the error in estimating the reverse pathway.

For a first generation of algorithmic approaches, the implementation can be considered a success, since the theory presented is an entirely new propostion. The details of the performance are also important however, and I would like to present an enhancement to this theory here, to be published in future work.

The theory begins with the POSH acceptance probability, as given in Chapter 4 Eq. 135:

Equation 161

$$\frac{acc_{13}}{acc_{31}} = \frac{p_3 T_{41}^{(N_I)}}{p_1 T_{23}^{(N_I)}}$$

where the forward transition matrices are computed as previously described in Eq. 137:

Equation 162

$$T_{i,j}^{(N_I)} = \prod_{k=1}^{N_I} t_{k-1,k}^{(i,j)}$$

If we introduce the simple idea that each inner step be generated using a hybrid Monte Carlo step, the transition probability at each (forward step) step (using the Metropolis criterion) is:

Equation 163

$$t_{k-1,k}^{(2,3)} = \delta(\mathbf{q}_{k-1}^{S} - \mathbf{q}_{k-1}^{T}) acc_{k-1,k}^{(2,3)} + \delta(\mathbf{q}_{k-1}^{S} - \mathbf{q}_{k-1})(1 - acc_{k-1,k}^{(2,3)})$$

where the acceptance probability is given by Eq. 29 of Chapter 1:

Equation 164

$$acc_{k-l,k}^{(2,3)} = \min\left(1, \exp\left[-\beta(H_{k-1}^{T} - H_{k-1})\right]\right)$$

where H_{k-1} and H_{k-1}^{T} are the initial and trial Hamiltonians, respectively, and are generated using the Hybrid Monte Carlo procedure described in Chapter 1. The salient feature of the HMC approach is that the size of the timestep can be arbitrarily controlled to such that Eq. 164 is nearly always unity: This assumption yields the following estimates of the forward and reverse transition probabilities:

Equation 165

$$T_{23}^{(N_I)} = \prod_{k=1}^{N_I} t_{k-1,k}^{(2,3)} \approx 1$$
$$T_{41}^{(N_I)} = \prod_{k=1}^{N_I} t_{k-1,k}^{(4,1)} \approx 1$$

where the value of the reverse transition probability can be estimated to be unity without the need to compute a reverse trajectory. It should be noted that the theory still asserts the existence of the reverse pathway, which has been demonstrated in the work of Chapter 4. This simple assumption immediately solves two of the major limitations of the original sampling method. It alleviates the need for computing the reverse trajectory, which was thought to be the major source of numeric error. It also provides a factor of 2 improvement in the efficiency of the sampling. An additional improvement which is not directly addressed is in the greatly reduced coordinate updates when constructing the reverse pathway that requires a significant amount of coordinate transformations and bookkeeping of coordinate states, while HMC requires only the bookkeeping of the Cartesian array in the forward pathway for the inner loop steps.

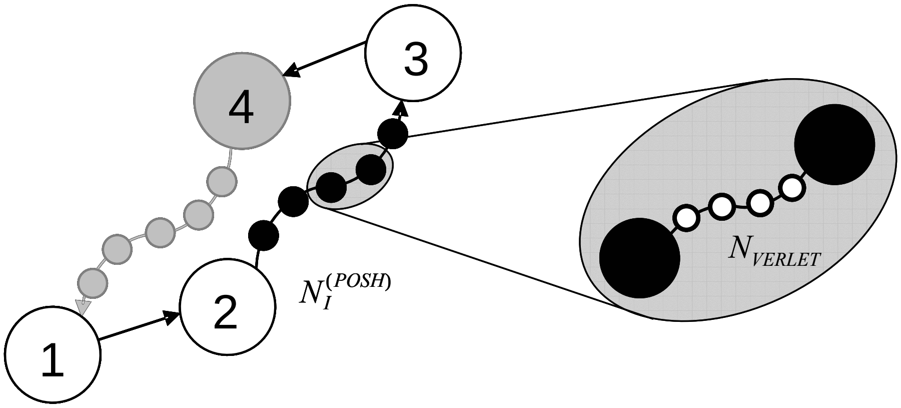


Figure 5.5 – POSH Hybrid Monte Carlo. Each inner step is computed using a Hybrid Monte Carlo move, consisting of N_{VERLET} molecular dynamics moves. Eq. 165 permits an estimate of the reverse transition probability that does not require the reverse pathway to be constructed.

There are other factors to consider, but this approach appears to have substantial promise. It appears to combine the basin hopping trial moves of the POSH formulation with the more natural annealing process of the HMC procedure. Initial tests have shown to be much more robust in parameter space. The sampling appears also to have high ergodicity, as has been shown with preliminary studies with Met-Enkephalin *in vacuo*.

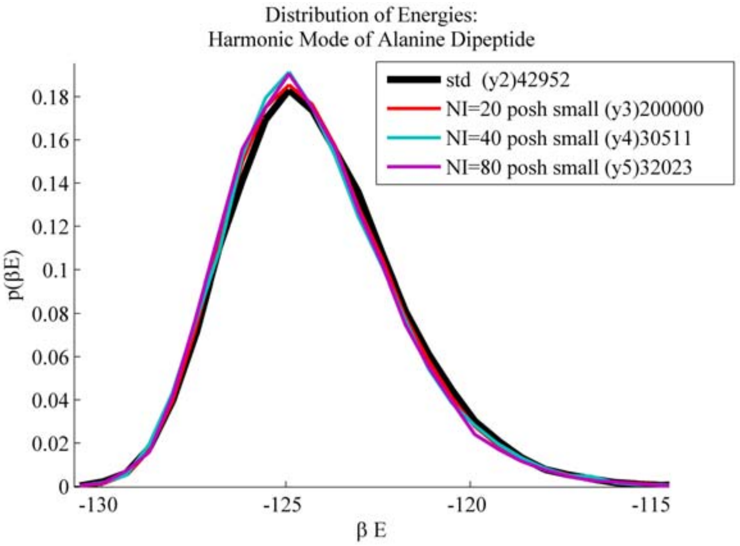


Figure 5.6 – Distribution of Energies using POSH-HMC for N_I =20,40,80. The deviations from the mean value are 0.02σ, 0.04σ, and 0.06σ, respectively. Compare to an expected deviation of 0.1σ using standard POSH at N_I=20.

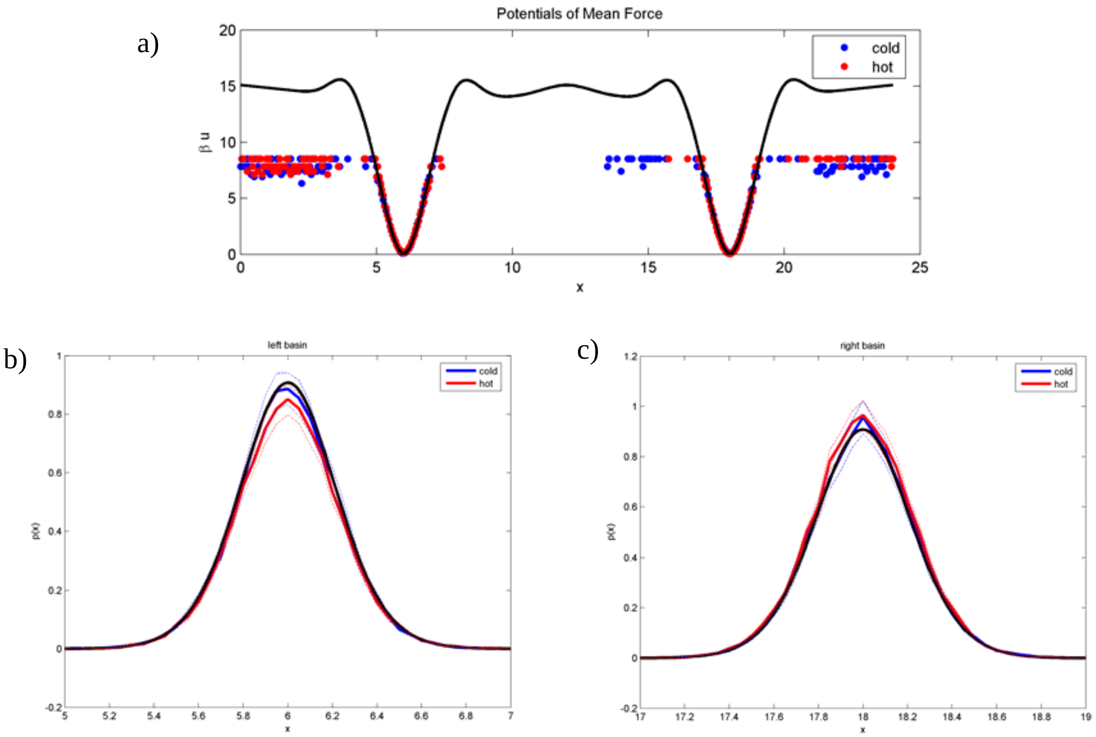


Figure 5.7 – Comparison of Replica Exchange protocols for multiple temperature POSH sampling and single temperature sampling shows that multiple temperature POSH sampling and replica exchange. ‘Hot’ replica is run at high temperature ($100T_0$) in the outer loops, while the ‘cold’ replica runs the inner loop at room temperature. Exchange rates are improved from 7% to 10% for the simple system shown here. Top Figure 5.is the 1D 2 basin potential with sampled data points, and the basin distributions are shown in the lower figures.

Parallel Sampling Methodology

If the notion may be advanced that constructing Monte Carlo acceptance criterion using the POSH formalism is a valid approach, then a framework for generating a variety of novel methods is possible. A simple extension of the POSH sampling framework is to establish different temperatures for the inner and outer loops. If the inner loop is sampled according to a low temperature, we can accept the trial move with a high temperature criterion. This has the practical consequence of generating a distribution of states in the high temperature ensemble that is nonetheless biased towards low temperature states. If a high temperature replica is generated using this procedure, a simple replica exchange protocol can be generated that has improved acceptance rates.

Continued Theoretical Work

It appears as though an interesting class of sampling approaches may be emerging from some of the simple ideas brought forth. It would be interesting to consider more applications in this direction, which may include nonequilibrium ideas, applications to free energy, and further advances with regard to replica exchange protocols.

Conclusions

The main goal of the thesis work was to incorporate a functionality that allowed for the sampling of proteins using a physics based model, and to generate distributions of states for which quantities like free energy and populations can be computed as meaningful quantities. The hope is threefold: 1) That the current methodologies be used to study a wide variety of proteins, 2) That the current extensible design of the code is accessible enough that it may be modified to include different degrees of freedom and novel geometric algorithms and 3) That some of the ideas in the Monte Carlo strategies be of use to the larger community of computational science.

References

(1) Dill, K.; Bromberg, S.; Yue, K.; Fiebig, K.; Yee, D.; Thomas, P.; Chan, H. *Protein Science* **1995**, *4*, 561.

(2) Dill, K.; Chan, H. *Nature Structural Biology* **1997**, *4*, 10.

(3) Dill, K. *Biochemistry* **1990**, *29*, 7133.

(4) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A.; Teller, E. *J Chem Phys* **1953**, *21*, 1087.

(5) D Frenkel, B. S. *Understanding Molecular Simulation: From Algorithms to Applications*; Academic Press: Boston, 2002.

(6) Kalos, M., Whitlock O. *Monte Carlo Methods - Volume I: Basics*, 1986.

(7) Allen, M. P., Tildesley. D.J. *Computer Simulation of Liquids*; Oxford University Press: New York, 1987.

(8) Van Kampen, N. G. *Stochastic Processes in Physics and Chemistry*Amsterdam, 1992.

(9) Manousiouthakis, V. I.; Deem, M. W. *J Chem Phys* **1999**, *110*, 2753.

(10) Hastings, W. K. *Biometrika* **1970**, *57*, 97.

(11) Rosenbluth, M. N.; Rosenbluth, A. W. *J Chem Phys* **1955**, *23*, 356.

(12) Rosenbluth, M. N.; Rosenbluth, A. W. *J Chem Phys* **1954**, *22*, 881.

(13) Frenkel, D.; Siepmann, J. I. *Molecular Physics* **1992**, *75*, 59.

(14) Bernacki, K.; Hetenyi, B.; Berne, B. J. *Journal of Chemical Physics* **2004**, *121*, 44.

(15) Hetenyi, B.; Bernacki, K.; Berne, B. J. *Journal of Chemical Physics* **2002**, *117*, 8203.

(16) Gelb, L. *The Journal of Chemical Physics* **2003**, *118*, 7747.

(17) Duane, S.; Kennedy, A.; Pendleton, B.; Roweth, D. *Physics Letters B* **1987**, *195*, 216.

(18) Box, G.; Muller, M. *Annals of Mathematical Statistics* **1958**, *29*, 610.

(19) Roux, B. *Implicit Solvent Models*; Marcel Dekker: New York City, 2001.

(20) Ghosh, A.; Rapp, C. S.; Friesner, R. A. *J. Phys. Chem. B.* **1998**, *102*, 10983.

(21) Deem, M. W. *Mol. Phys.* **1999**, *97*, 559.

(22) Dunbrack, R. L., Jr.; Karplus, M. *Nat. Struct. Biol.* **1994**, *1*, 334.

(23) Dunbrack, R. L., Jr.; Cohen, F. E. *Protein Sci.* **1997**, *6*, 1661.

(24) Xiang, Z.; Honig, B. *J. Mol. Biol.* **2001**, *311*, 421.

(25) Kuhlman, B.; Baker, D. *Proc. Natl. Acad. Sci. U.S.A.* **2000**, *97*, 10383.

(26) Jiang, L.; Kuhlman, B.; Kortemme, T.; Baker, D. *Proteins* **2005**, *58*, 893.

(27) Fiser, A.; Do, R. K.; Sali, A. *Protein Sci.* **2000**, *9*, 1753.

(28) Fiser, A.; Sali, A. *Methods Enzymol.* **2003**, *374*, 461.

(29) Sherman, W.; Day, T. J.; Jacobson, M.; Friesner, R. A.; Farid, R. *J. Med. Chem.* **2006**, *49*, 534.

(30) Meiller, J.; Baker, D. *Proteins* **2006**, *65*, 538.

(31) Ferrari, A. M.; Wei, B.; Constantino, L.; Shoichet, B. K. *J. Med. Chem.* **2004**, *47*, 5076.

(32) Voigt, C. A.; Gordon, D. B.; Mayo, S. L. *J. Mol. Biol.* **2000**, *299*, 789.

(33) Jain, T.; Cerutti, D. S.; McCammon, J. A. *Protein Sci.* **2006**, *15*, 2029.

(34) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. *J. Mol. Biol.* **1982**, *161*, 269.

(35) Schlick, T. *Molecular Modeling and Simulation*; Springer-Verlag, 2002.

(36) Mobley, D. *J. Chem. Theory Comput.* **2007**, *3*, 1231.

(37) Mobley, D.; Graves, A.; Chodera, J. D.; McReynolds, A.; Shoichet, B. K.; Dill, K. A. *J. Mol. Biol.* **2007**, *371*, 1118.

(38) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087.

(39) Rosenbluth, M. N.; Rosenbluth, A. W. *J. Chem. Phys.* **1955**, *23*, 356.

(40) Deem, M. W. *J.Chem. Phys.* **1999**, *111*, 6625.

(41) Dinner, A. R. *J. Comput. Chem.* **2000**, *21*, 1132.

(42) Ulmschneider, J. P.; Jorgensen, W. L. *J. Am. Chem. Soc.* **2004**, *126*, 1849.

(43) Li, Z. Q.; Scheraga, H. A. *Proc. Natl. Acad. Sci. U. S. A.* **1987**, *84*, 6611.

(44) Sorin, E. J.; Engelhardt, M. A.; Herschlag, D.; Pande, V. S. *J. Mol. Biol.* **2002**, *317*, 493.

(45) Pande, V. S.; Baker, I.; Chapman, J.; Elmer, S. P.; Khaliq, S.; Larson, S. M.; Rhee, Y. M.; Shirts, M. R.; Snow, C. D.; Sorin, E. J.; Zagrovic, B. *Biopolymers* **2003**, *68*, 91.

(46) Felts, A. K.; Gallicchio, E.; Wallqvist, A.; Levy, R. M. *Proteins* **2002**, *48*, 404.

(47) W. Clark Still, A. T., Ronald C. Hawley, and; Hendrickson, T. *J. Am. Chem. Soc.* **1990**, *112*, 6127.

(48) Gallicchio, E.; Zhang, L. Y.; Levy, R. M. *J. Comput. Chem.* **2002**, *23*, 517.

(49) Kaminski, G. A.; Friesner, R. A.; Tirado-Rives, J.; Jorgensen, W. L. *J. Phys. Chem. B.* **2001**, *105*, 6474.

(50) Jacobson, M. P.; Pincus, D. L.; Rapp, C. S.; Day, T. J.; Honig, B.; Shaw, D. E.; Friesner, R. A. *Proteins* **2004**, *55*, 351.

(51) Jacobson, M. P.; Friesner, R. A.; Xiang, Z.; Honig, B. *J. Mol. Biol.* **2002**, *320*, 597.

(52) Tuckerman, M.; Berne, B. J. *J. Chem. Phys.* **1990**, *94*, 1465.

(53) Tuckerman, M.; Berne, B. J.; G.J., M. *J. Chem. Phys.* **1992**, *97*, 1990.

(54) Hetenyi, B.; Bernacki, K.; Berne, B. J. *J. Chem. Phys.* **2002**, *117*, 8203.

(55) Michel, J.; Taylor, R.; Essex, J. *J. Chem. Theory Comput.* **2006**, *2*, 732.

(56) Yu, Z.; Jacobson, M. P.; Friesner, R. A. *J. Comp. Chem.* **2005**, *27*, 72.

(57) Jacobson, M. *J. Phys. Chem. B* **2004**, *108*, 6643.

(58) Verlet, L. *Phys. Rev.* **1968**, *165*, 201.

(59) Weeks, J. D.; Chandler, D.; Andersen, H. C. *J. Chem. Phys.* **1971**, *55*, 5422.

(60) Bernacki, K.; Hetenyi, B.; Berne, B. J. *J. Chem. Phys.* **2004**, *121*, 44.

(61) Gelb, L. D. *J. Chem. Phys.* **2003**, *118*, 7747.

(62) Zhu, K.; Friesner, R. A.; Jacobson, M. P. *J. Comp. Chem.* **2006**.

(63) Chen, B.; Siepmann, J. I. *Theor. Chem. Acc.* **1999**, *103*, 87.

(64) Wei Yang, R. B.-P., Martin Karplus. *J. Chem. Phys.* **2004**, *120*.

(65) S. Shapiro, M. B. W., and H. J. Chen. **1968**, *63*, 1343.

(66) S. Shapiro and M. B. Wilk. *Biometrika* **1965**, *52*, 591.

(67) Arevalo, J. H.; Hassig, C. A.; Stura, E. A.; Sims, M. J.; Taussig, M. J.; Wilson, I. A. *J. Mol. Biol.* **1994**, *241*, 663.

(68) Arevalo, J. H.; Stura, E. A.; Taussig, M. J.; Wilson, I. A. *J. Mol. Biol.* **1993**, *231*, 103.

(69) Knight, Z. A.; Gonzalez, B.; Feldman, M. E.; Zunder, E. R.; Goldenberg, D. D.; Williams, O.; Loewith, R.; Stokoe, D.; Balla, A.; Toth, B.; Balla, T.; Weiss, W. A.; Williams, R. L.; Shokat, K. M. *Cell* **2006**, *125*, 733.

(70) Roe, D. R.; Okur, A.; Wickstrom, L.; Hornak, V.; Simmerling, C. *J. Phys. Chem. B* **2007**, *111*, 1846.

(71) Senda, M.; Senda, T.; Ogi, T.; Kidokoro, S.; Stihle, R.; Boroni, E.; Hennig, M. *Acta. Cryst.* **2002**, *58*, C278.

(72) Shiau, A. K.; Barstad, D.; Loria, P. M.; Cheng, L.; Kushner, P. J.; Agard, D. A.; Greene, G. L. *Cell* **1998**, *95*, 927.

(73) Pike, A. C.; Brzozowski, A. M.; Walton, J.; Hubbard, R. E.; Bonn, T.; Gustafsson, J. A.; Carlquist, M. *Biochem. Soc. Trans.* **2000**, *28*, 396.

(74) Gampe, R. T., Jr.; Montana, V. G.; Lambert, M. H.; Miller, A. B.; Bledsoe, R. K.; Milburn, M. V.; Kliewer, S. A.; Willson, T. M.; Xu, H. E. *Mol. Cell* **2000**, *5*, 545.

(75) Nolte, R. T.; Wisely, G. B.; Westin, S.; Cobb, J. E.; Lambert, M. H.; Kurokawa, R.; Rosenfeld, M. G.; Willson, T. M.; Glass, C. K.; Milburn, M. V. *Nature* **1998**, *395*, 137.

(76) Meijer, L.; Thunnissen, A. M.; White, A. W.; Garnier, M.; Nikolic, M.; Tsai, L. H.; Walter, J.; Cleverley, K. E.; Salinas, P. C.; Wu, Y. Z.; Biernat, J.; Mandelkow, E. M.; Kim, S. H.; Pettit, G. R. *Chem. Biol.* **2000**, *7*, 51.

(77) Bourne, Y.; Watson, M. H.; Hickey, M. J.; Holmes, W.; Rocque, W.; Reed, S. I.; Tainer, J. A. *Cell* **1996**, *84*, 863.

(78) Groban, E. S.; Narayanan, A.; Jacobson, M. P. *PLoS Comput. Biol.* **2006**, *2*, e32.

(79) Coutsias, E. A.; Seok, C. L.; Jacobson, M. P.; Dill, K. A. *J. Comp. Chem.* **2004**, *25*, 510.

(80) Go, N.; Scheraga, H. A. *Macromolecules* **1969**, *3*, 178.

(81) Dodd, L. R.; Boone, T. D.; Theodorou, D. N. *Mol. Phys.* **1993**, *78*, 961.

(82) Wong, S.; Jacobson, M. P. *Proteins* **2008**, *71*, 153.

(83) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. *J. Comput. Chem.* **2004**, *25*, 1605.

(84) Shell, M.; Debenedetti, P.; Panagiotopoulos, A. *The Journal of Chemical Physics* **2003**, *119*, 9406.

(85) Wedemeyer, W. J.; Scheraga, H. A. *Biophysical Journal* **2000**, *78*, 333a.

(86) Wedemeyer, W. J.; Scheraga, H. A. *J Comp Chem* **1999**, *20*, 819.

(87) Dinner, A. *Journal of Computational Chemistry* **2000**, *21*, 1132.

(88) Coutsias, E. A.; Seok, C. L.; Jacobson, M. P.; Dill, K. A. *Journal of Computational Chemistry* **2004**, *25*, 510.

(89) Dodd, L. R.; Boone, T. D.; Theodorou, D. N. *Molecular Physics* **1993**, *78*, 961.

(90) Ho, B.; Coutsias, E.; Seok, C.; Dill, K. *Protein Science* **2005**, *14*, 1011.

(91) Ho, B. K.; Coutsias, E. A.; Seok, C.; Dill, K. A. *Protein Sci* **2005**, *14*, 1011.

(92) Panagiotopoulos, A. Z.; Quirke, N.; Stapleton, M.; Tildesley, D. J. *Molecular Physics* **1987**, *63*, 527.

(93) Panagiotopoulos, A. Z. *Molecular Physics* **2002**, *100*, 237.

(94) De Pablo, J. J.; Prausnitz, J. M. *Fluid Phase Equilibria* **1989**, *53*, 177.

(95) Meirovitch, H. *J Phys A* **1982**, *15*, L735.

(96) Meirovitch, H. *Proceedings of the National Academy of Sciences of the United States of America* **2004**, *101*, 9235.

(97) Madras, N.; Sokal, A. *Journal of Statistical Physics* **1988**, *50*, 109.

(98) Chen, B.; Siepmann, J. I. *Theoretical Chemistry Accounts* **1999**, *103*, 87.

(99) Chen, B.; Xing, J.; Siepmann, J. I. *J Phys Chem B* **2000**, *104*, 2391.

(100) Li, Z. Q.; Scheraga, H. A. *Proceedings of the National Academy of Sciences of the United States of America* **1987**, *84*, 6611.

(101) David, L.; Luo, R.; Gilson, M. K. *J Comput Aided Mol Des* **2001**, *15*, 157.

(102) Killian, B. J.; Yundenfreund Kravitz, J.; Gilson, M. K. *J Chem Phys* **2007**, *127*, 024107.

(103) Opps, S.; Schofield, J. *Physical Review E* **2001**, *63*, 56701.

(104) Brown, S.; Head-Gordon, T. *J Comp Chem* **2003**, *24*, 68.

(105) Frantz, D. D.; Freeman, D. L.; Doll, J. D. *J Chem Phys* **1990**, *93*, 2769.

(106) Xu, H.; Berne, B. J. *J Chem Phys* **1999**, *110*, 10299.

(107) Zhou, R. H. *J Chem Phys* **1997**, *107*, 9185.

(108) Huber, G.; McCammon, J. *Physical Review E* **1997**, *55*, 4822.

(109) Sugita, Y.; Okamoto, Y. *Chemical Physics Letters* **1999**, *314*, 141.

(110) Lyman, E.; Ytreberg, F.; Zuckerman, D. *Physical Review Letters* **2006**, *96*, 28105.

(111) Roitberg, A.; Okur, A.; Simmerling, C. *Journal of physical chemistry. B, Condensed matter, materials, surfaces, interfaces, & biophysical chemistry* **2007**, *111*, 2415.

(112) Bandyopadhyay, P. *The Journal of Chemical Physics* **2008**, *128*, 134103.

(113) Gordon, M.; Freitag, M.; Bandyopadhyay, P.; Jensen, J.; Kairys, V.; Stevens, W. *JOURNAL OF PHYSICAL CHEMISTRY A* **2001**, *105*, 293.

(114) Fiser, A.; Sali, A. *Methods Enzymol* **2003**, *374*, 461.

(115) Rohl, C. A.; Strauss, C. E.; Misura, K. M.; Baker, D. *Methods Enzymol* **2004**, *383*, 66.

(116) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. *J Mol Biol* **1982**, *161*, 269.

(117) Foreman, K. W.; Phillips, A. T.; Rosen, J. B.; Dill, K. A. *J Comp Chem* **1999**, *20*, 1527.

(118) Rosso, L.; Mináry, P.; Zhu, Z.; Tuckerman, M. *The Journal of Chemical Physics* **2002**, *116*, 4389.

(119) Rosso, L.; Tuckerman, M. *Molecular Simulation* **2002**, *28*, 91.

(120) Darve, E.; Wilson, M.; Pohorille, A. *Molecular Simulation* **2002**, *28*, 113.

(121) Nilmeier, J.; Jacobson, M. *Journal of Chemical Theory and Computation, Submitted.*

(122) Onuchic, J.; Luthey-Schulten, Z.; Wolynes, P. *Annual Reviews in Physical Chemistry* **1997**, *48*, 545.

(123) Dellago, C.; Bolhuis, P.; Csajka, F.; Chandler, D. *statistics* **1964**, *10*, 15.

(124) Valleau, J.; Whittington, S. *Journal of Computational Physics* **1977**, *24*, 150.

(125) Dellago, C.; Bolhuis, P.; Csajka, F.; Chandler, D. *The Journal of Chemical Physics* **1998**, *108*, 1964.

(126) Bolhuis, P.; Dellago, C.; Geissler, P.; Chandler, D. *JOURNAL OF PHYSICS CONDENSED MATTER* **2000**, *12*, 147.

(127) Singhal, N.; Snow, C.; Pande, V. *The Journal of Chemical Physics* **2004**, *121*, 415.

(128) Cao, J.; Berne, B. J. *The Journal of Chemical Physics* **1990**, *92*, 1980.

(129) Zhou, R.; Berne, B. *The Journal of Chemical Physics* **1997**, *107*, 9185.

(130) Wong, S. E.; Bernacki, K.; Jacobson, M. *J Phys Chem B* **2005**, *109*, 5249.

(131) Smart, J., McCammon, JA. *Biopolymers* **1999**, *49*, 225.

(132) Rizzo, R. C.; Jorgensen, W. L. *Journal of the American Chemical Society* **1999**, *121*, 4827.

(133) Groban, E. S.; Narayanan, A.; Jacobson, M. P. *PLoS Comput Biol* **2006**, *2*, e32.

(134) Haliloglu, T.; Bahar, I.; Erman, B. *Physical Review Letters* **1997**, *79*, 3090.

(135) Atilgan, A.; Durell, S.; Jernigan, R.; Demirel, M.; Keskin, O.; Bahar, I. *Biophysical Journal* **2001**, *80*, 505.

(136) Wong, S.; Jacobson, M. P. *Proteins* **2008**, *71*, 153.

Publishing Agreement

It is the policy of the University to encourage the distribution of all theses and dissertations. Copies of all UCSF theses and dissertations will be routed to the library via the Graduate Division. The library will make all theses and dissertations accessible via the Graduate Division. The library will make all theses and dissertations accessible to the public and will preserve these to the best of their abilities, in perpetuity.

I hereby grant permission to the Graduate Division of the University of California, San Francisco to release copies of my thesis or dissertation to the Campus Library to provide access and preservation, in whole or in part, in perpetuity.

Jerome Nilmeier

Jerome P. Nilmeier

09 Sep 2008

Date