Systematic Coarse-Graining in Molecular Simulations using Relative Entropy and Generalized Ensembles

by

© Aidan Tremblett

A Thesis submitted to the School of Graduate Studies in
partial fulfillment of the requirements for the degree of

Master of Science in Physics

Department of Physics and Physical Oceanography

Memorial University of Newfoundland

St. John's, Newfoundland

August 2019

Abstract

Computer simulations have become a powerful tool for studying the structure, dynamics, or other characteristics of a wide variety of physical systems. The goal of coarse-grained (CG) models is to simplify the representation of the physical system while still maintaining enough information to capture the desired properties of the system. A main challenge in the development of CG models is determining the potential energy function, U_{CG}, which often depends on a large number of unknown model parameters, $\overline{\lambda}$. Different methods for determining these model parameters have been proposed, (potential of mean force, multi-scale coarse-graining), but they rely on determining quantities, such as free energies, that are computationally challenging to calculate.

Here we develop a systematic method to determine the optimal parameters for coarse-grained models of molecular systems, using the relative entropy, S_{rel}, as a metric to compare a target ensemble to an ensemble generated from a CG model. The relative entropy depends on the free energy, and a novel approach for determining the free energy was developed, which used a generalized ensemble approach to simulate the joint probability distribution, $p(\overline{r}, \overline{\lambda})$, where $\overline{r}$ is a chain conformation. The generalized ensemble Monte Carlo simulation allowed the model parameters to be dynamic, which means they are allowed to change during the simulation. These simulations allow for the free energies, $F_{\mathrm{CG}}(\lambda)$, to be obtained directly from the marginal probability distribution, $p(\overline{\lambda})$, during the simulation. The relative entropy, $S_{\text{rel}}(\lambda)$, was calculated and minimized with respect to the CG model parameters in order to obtain the optimal model parameters.

The systematic method was applied to an existing CG model for protein folding that was modified to include a new potential energy term that contained either 13 or 91 unknown model parameters. The method was used to systematically determined the optimal model parameters that allowed a protein to fold to its native structure. The relative entropy was calculated for two target ensembles, the experimentally determined single native structure, and the set of configurations from an all-atom simulation. It was found that the potential energy function with 91 unknown parameters converged to the optimal parameter set faster than the potential energy function with 13 unknown parameters. The optimal parameter set for the 13 model parameters was not able to fully capture the folding of the protein, while the 91 model parameter set was able to capture the folding behaviour. Furthermore, the optimal CG model parameter set that was found using the experimentally determined native structure as a target for the relative entropy minimization gave better results than the all-atom target ensemble. This is likely due to the set of configurations for the all-atom target ensemble being dominated by the unfolded state instead of a folded state.

Acknowledgements

I would sincerely like to thank my supervisor, Dr. Stefan Wallin, for the guidance and support with everything throughout this project. I would also like to thank Memorial University of Newfoundland and the Department of Physics and Physical Oceanography for the opportunity and funding to complete this project.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computer simulations are increasingly being used to study the structure and dynamics of physical systems in a variety of fields, with a significant amount of work done in physics, biology, and chemistry to simulate various molecular systems [1]. Despite technological advances, computer simulations are still limited by the processing power of the computer hardware being used. To overcome this limitation, coarse-graining methods were introduced to simplify the computer models while still maintaining enough information to capture important properties of the physical system. Coarse-grained simulations, as opposed to fine-grained or all-atom simulations, allow larger or more complex molecular systems to be studied for longer time-scales.

However, one issue that can occur in coarse-grained molecular simulations is the inability to accurately reproduce the fine-grained simulations or experimental results under the same thermodynamic conditions. This issue arises when the chosen CG model for representing the physical system does not contain enough detail, and thus, is unable to capture the important properties of the system. To overcome this issue, the correct or optimal CG model must be determined. Another common problem for a large class of molecular systems is called the multiple-minima problem, where the potential energy landscape has a large number of local minima separated by high energy barriers. This causes a problem in computer simulations because the system can get trapped in a local minimum. However, this problem can be overcome by applying a generalized ensemble approach to the computer simulation, which allows the system to sample all states uniformly, regardless of the energy landscape.

The goal of this thesis was to develop a novel systematic method to determine optimal model parameters for coarse-grained models of molecular systems. This was accomplished by using the relative entropy and generalized ensemble methods to allow for efficient parallel exploration of parameter space. The systematic method was applied to an existing CG model used for studying protein folding. Using various optimization methods, the method was able to recapture the correct values for known model parameters. Furthermore, the CG model was modified and the systematic CG method was applied to determine unknown model parameters for a real protein by comparing the simulation results with the experimentally determined native structure.

1.1 Coarse-Graining

Computational simulations are used extensively to study the structure and dynamics of a wide range of physical systems. One of the main challenges of simulating physical systems is generating a computer model that accurately represents the physical system of interest. This can often be accomplished by modelling systems classically, at full atomistic detail, and determining all forces and interactions involved. These models are called all-atom or first-principle models. However, due to the complexity of the all-atom models, they are limited to small systems and short time scales.

Methods were developed that simplify the representation of the physical system while still maintaining enough information to enable simulations to capture the desired system characteristics. These so called coarse-grained methods (CG) allow for larger systems to be simulated on longer time scales. The granularity, or degree of simplification, varies greatly depending on what characteristic of the system are being studied. Typically, CG models are developed to either reproduce average structural properties obtained in fine-grained, all-atom simulations (*bottom-up approach*), or to match experimentally determined thermodynamic properties (*top down approach*) [2, 3]. One of the main issues encountered in the development of coarse-grained models is the determination of the CG interaction potential energy, U_{CG}. The interaction potential energy of a CG model must be capable of reproducing the behaviour of the "target" or first-principles model which has a known interaction potential. This is important in CG simulations, as it provides a way to compare results from the CG model with results from a target model (typically an all-atom simulation).

Some of the systematic approaches by which coarse-grained models are developed [4] include a structure-based approach [5], knowledge-based statistical approach [6], and a physics-based or force matching approach [7]. Structure-based approaches, also called Gō-type models, use specific force field approximations that only account for interaction patterns that allow the CG model to form known structures. Knowledge-based statistical approaches use statistical analysis of information from experimentally determined structures to determine the interaction potentials for the CG model. In physics-based methods, the goal is to derive an equation for the coarse-grained potential energy function that enables the CG model to reproduce a target radial distribution function (RDF) or a target force distribution [3]. In practice, a large number of potential energy functions will allow a CG model to match a physical system. Methods to determine the potentials include the Inverse Monte Calro (IMC) method, which uses an iterative scheme to correct a guess potential [8], or the Force Matching (FM) methods, which use first-principle calculations to fit the CG potentials to the atomic forces. Other methods include the iterative Boltzmann inversion (IBI) scheme, which can determine effective potentials by using a set of correlation functions [9].

Additional methods for determining the CG potential energy include the potential of mean force (PMF) method [10], the Relative Entropy formalism by Shell [11], or the Multi-Scale Coarse-Graining (MSCG) method by Izvekov and Voth [2]. In the PMF method, the optimal CG model can be determined by calculating the potential of mean force (PMF) of the first principles model over the CG model degrees of freedom [10]. The Relative Entropy method is a general case of the IMC method, where Shell used the relative entropy as a measure of the amount of information lost due to coarse-graining. By minimizing the entropy, the optimal CG force-field can be obtained. The MSCG method is an extension of the FM method, where Molecular Dynamics (MD) all-atom simulations are used as reference in determining the CG potentials [4]. This method allows for the atomistic-level forces present in all-atom simulations to be "propagated upward" in scale to the coarse-grained level [2]. A variety of these methods have been compiled together in the Versatile Object-Oriented Toolkit for Coarse-graining Applications (VOTCA) software package, which provides a way to compare CG potentials obtained by various methods [12].

By using information from all-atom simulations in the systematic development of CG potentials, the CG model will reproduce system properties that are observed in all-atom simulations under the same thermodynamic conditions. This is advantageous because the CG model will allow for simulations of larger systems and longer time-scales, while recapturing important system properties.

1.2 Relative Entropy

The relative entropy, or Kullback-Leibler divergence, gives a statistical measure of the difference between two probability distributions, $p_1(r)$ and $p_2(r)$. It is given by the formula,

$$S_{\mathrm{rel}} = \sum_{r \in R} p_1(r) \ln \frac{p_1(r)}{p_2(r)} \tag{1.1}$$

where r is a discrete variable in the set R. The relative entropy has the important property of always being non-negative, and $S_{rel} = 0$ if and only if $p_1 = p_2$ everywhere [13].

Shell used the relative entropy in a CG protein simulation as a method to compare the probability distribution of a thermodynamic CG model ensemble, $p_{\mathrm{M}}(r)$, and some existing all-atom (AA) target ensemble, $p_{\mathrm{T}}(r)$ [11]. In this application, the properties of the relative entropy dictate that the function is minimized when the model ensemble best matches the target ensemble.

One alteration must be made in order to account for the case where the model system has fewer degrees of freedom than the target system. In this case, a mapping function, M is required to allow for the set of coordinates of any configuration in the target ensemble, r_{T}, to be mapped to a set of coordinates in the model ensemble, r_{M}. The relationship is given as

$$r_{\mathrm{M}} = M(r_{\mathrm{T}}). \tag{1.2}$$

However, since the model has fewer degrees of freedom, it is possible that multiple configurations in the target ensemble will map to the same model configuration. A measure of that degeneracy must be included when considering the probability of generating a configuration in the model ensemble. The complexities of the mapping can be absorbed into a single term, $\langle S_{\mathrm{map}} \rangle_{\mathrm{T}}$, which is the average entropy that occurs from the target to model mapping. Then, the relative entropy between a CG model ensemble and some target ensemble is given by

$$S_{\mathrm{rel}} = \sum_r p_{\mathrm{T}}(r) \ln \frac{p_{\mathrm{T}}(r)}{p_{\mathrm{M}}(M(r))} + \langle S_{\mathrm{map}} \rangle_{\mathrm{T}} \tag{1.3}$$

where the summation is over all possible configurations, r, in the target ensemble.

In the canonical ensemble, the probability distribution can be written in terms of the partition function, Z, and the potential energy, U, such that $p(r) = \frac{1}{Z} e^{-\beta U(r)}$, which gives

$$S_{\mathrm{rel}} = \sum_r p_{\mathrm{T}}(r) \ln \left[\frac{Z_{\mathrm{CG}}}{Z_{\mathrm{T}}} e^{\beta(U_{\mathrm{CG}} - U_{\mathrm{T}})} \right] + \langle S_{\mathrm{map}} \rangle_{\mathrm{T}}. \tag{1.4}$$

Using the relation between the Helmholtz free energy and the partition function, $F = -k_b T \ln Z$, the relative entropy can be written

$$S_{\mathrm{rel}} = \beta \sum_i p_{\mathrm{T}}(r) \big(U_{\mathrm{CG}} - U_{\mathrm{T}} \big) + \ln \left[e^{-\beta F_{\mathrm{CG}} + \beta F_{\mathrm{T}}} \right] \times \sum_r p_{\mathrm{T}}(r) + \langle S_{\mathrm{map}} \rangle_{\mathrm{T}}. \tag{1.5}$$

Simplifying further, the relative entropy equation is written as

$$S_{\mathrm{rel}} = \beta \langle U_{\mathrm{CG}} - U_{\mathrm{T}} \rangle_{\mathrm{T}} - \beta \big(F_{\mathrm{CG}} - F_{\mathrm{T}} \big) + \langle S_{\mathrm{map}} \rangle_{\mathrm{T}} \tag{1.6}$$

where $\beta = 1/k_{\mathrm{B}}T$, U is the potential energy, F is the configurational part of the Helmholtz free energy, and the brackets $\langle \rangle_{\mathrm{T}}$ indicates an average in the target ensemble. The mapping entropy is independent of the properties of the model ensemble, thus it will only affect the relative entropy by shifting the value by a constant. The exact value of the relative entropy is not required, so the mapping term is ignored.

1.3 Metropolis Monte Carlo Simulations

The goal of a computer simulation is to allow the model system to move and sample a variety of geometrical configurations. Computer simulations fit into one of two categories: Molecular Dynamics (MD) or Monte Carlo (MC) [4]. In MD simulations, the new configurations are generated by updating the position and momenta of each atom using Newton's equations of motion over a small time-step. Monte Carlo (MC) method, used here, allows the system to randomly sample configurations by making small modifications and using a probability distribution to accept or reject the modification.

The MC method used here was developed by Metropolis *et al.* [14] as a general MC method for calculating thermodynamic properties of the system. The method generates N configurations, denoted $r_1, r_2, \ldots, r_N$, where the probability of finding the system with the configuration r_i is given by $p(r_i) \propto e^{-\beta U(r_i)}$. The equation for determining average system properties is given by

$$\langle A(r) \rangle = \frac{1}{M} \sum_{i=1}^{M} A(r_i) \tag{1.7}$$

where $A(r)$ is a configuration dependent property of the system, and M is the number of moves.

The system samples different configuration by attempting to make changes to the geometry, called trial moves, which are tested against an acceptance criteria before being accepted. Suppose the system is in the configuration, r_a, and is perturbed a small amount such that it has a new configuration, r_b. The Metropolis algorithm uses an acceptance criteria such that the probability of sampling states is proportional to $\exp[-\beta U(r)]$.

The acceptance criteria for a trial move is based on the “detailed balance” equation. The transition probability, $W(a \to b)$, is defined as the probability that the system will move from configuration a to configuration b. The Metropolis algorithm states that the probability of a transition from a to b must be equal to the probability of a transition from b to a. The detailed balance equation below gives the sufficient but not necessary condition for satisfying the reversibility of the transition.

$$P(a)W(a \to b) = P(b)W(b \to a) \tag{1.8}$$

where the transition probability is related to the acceptance probability by $W(a \to b) = T(a \to b)P_{\text{acc}}(a \to b)$, where $T(a \to b)$ is a symmetric proposal probability.

It follows that

$$\frac{W(a \to b)}{W(b \to a)} = \frac{P(b)}{P(a)} = \exp[-\beta(U(b) - U(a))] = \exp[-\beta\Delta U] \tag{1.9}$$

where $\Delta U = U(b) - U(a)$.

The Metropolis method condition that satisfies Eq. 1.9 is given as

$$P_{\text{acc}}(a \to b) = \min\left(1, \frac{P(b)}{P(a)}\right) = \min\bigl(1, \exp[-\beta\Delta U]\bigr). \tag{1.10}$$

This means if $\Delta U < 0$, the trial move brings the system to a lower energy state, and the move is always accepted. However, if $\Delta U > 0$, the move is accepted with the probability $\exp[-\beta\Delta U]$. In practice, this is done by generating a uniform random number $\xi \in [0, 1]$ and accepting the move only if $\xi < \exp[-\beta\Delta U]$.

1.4 Generalized Ensemble Techniques

Although Metropolis Monte Carlo simulations are commonly used to study molecular systems, they often encounter a convergence problem at low temperature where the energy landscape has a large number of local minima that have high energy barriers. In this situation, the system will get trapped in local minimum, and only explore configurations in a small section of the energy landscape. This will lead to inaccurate calculations of physical quantities because the configurations will be correlated. This issue can be alleviated with the generalized ensemble (GE) method [15], which was derived from the standard Metropolis MC method.

The generalized ensemble approach employs a non-Boltzmann probability weight factor to each state such that the entire energy landscape may be sampled [16]. The goal of GE is to allow the system to sample rare or important states frequently, and enable the system to escape high energy barrier states. Three examples using the GE approach are the Multicanonical algorithm, $1/k$ sampling, and simulated tempering, all of which perform a simulation over an ensemble that is defined such that a chosen physical quantity obtains a uniform (noncanonical) distribution [15].

In GE simulations, the chosen quantity is called a control parameter, and is discretized into a set of evenly spaced values, A_n. Then, over the course of the simulation, the control parameter is allowed to change under some criteria, which results in the system taking on the new control parameter value. The advantage of the GE approach is that the global minimum energy state can be determined from one simulation, as the simulation is allowed to sample configurations at different values of the control parameter.

The simulated tempering method, originally developed by Marinari and Parisi in 1992 [17], is a GE method in which the temperature is the control parameter. Using the Metropolis MC method, the simulated tempering method allows the temperature control parameter to make a trial move, which would either increase or decrease the temperature of the system by a small amount. The acceptance criteria must be modified to account for the control parameter weight function, $w(\beta, s)$, which depends on the inverse temperature, $\beta = 1/k_BT$ and the current state, s. The Metropolis acceptance criteria for a control parameter trial move is given as

$$P((\beta, s) \to (\beta', s)) = \min\left(1, \frac{w(\beta', s)}{w(\beta, s)}\right) \tag{1.11}$$

where the weight function is given by

$$w(\beta, s) = \exp[-\beta U(s) + g(\beta)]. \tag{1.12}$$

The function $g(\beta)$ is a control parameter dependent function, where the optimal choice for this function is the free energy of the system, $g(\beta) = \beta F(\beta)$ [18]. By defining $g(\beta)$ in this way, it is possible to calculate the free energy throughout a simulated tempering simulation.

1.5 Free Energy Calculation

One of the most challenging quantities to estimate with precision in computer simulations is the free energy, F, as a function of global variables, such as temperature, T. The reason this is challenging is because the entropic factors require comprehensive sampling of the system in all possible states. Thus, calculating estimates of the free energy is often time consuming and computationally intensive. Methods, such as simulated tempering [17] or umbrella sampling [19], can be used to obtain better estimates for free energies by doing comprehensive sampling.

One way to overcome the computationally intensive process was to combine the information from all of the sampled states of the system at different thermodynamic conditions. Early methods to compute free energy estimates using this concept included one-sided exponential averaging (EXP)[20], and Bennett acceptance ratio method (BAR)[21]. Both of these gave better estimates, but did not make efficient use of all the data. Multiple histogram reweighting techniques were developed to include data from multiple states to calculate estimates [22]. These techniques provided significantly improved estimates for the free energy differences, and allowed for estimates at arbitrary thermodynamic states, which included states not sampled.

However, due to the limitations of using binned energy histograms in the multiple histogram reweighting techniques, a method called the multistate Bennett acceptance ratio estimator (MBAR) was developed. This method allows for the calculation of statistically optimal estimates for the free energy differences for arbitrary thermodynamic states using data from multiple thermodynamic states [23].

The result of interest from the MBAR method is an estimating equation for the dimensionless free energies, which is derived for configurations that are sampled with Boltzmann statistics. Starting with N_i uncorrelated samples from K different thermodynamic states, the configurations $\{\boldsymbol{x}_{in}\}_{n=1}^{n_i}$ from state i have the probability distribution

$$p_i(\boldsymbol{x}) = c^{-1} q_i(\boldsymbol{x}) \tag{1.13}$$

where $q_i(\boldsymbol{x})$ is the unnormalized density function, and c_i is the normalization constant. For Monte Carlo sampling, the unnormalized density function is the Boltzmann distribution, $q_i(\boldsymbol{x}) = \exp[-U_i(\boldsymbol{x})]$.

Following the derivation in Appendix A, the equation for the estimated free energies, $\hat{f}_i$ is found to be

$$f_i = -\ln \sum_{j=1}^{K} \sum_{n=1}^{N_j} \frac{e^{-U_i(\boldsymbol{x}_{jn})}}{\sum_{k=1}^{K} N_k e^{(f_k - U_k(\boldsymbol{x}_{jn}))}}. \tag{1.14}$$

This equation is a self-consistent solution for $\hat{f}_i$, and can be solved using an iterative approach by using the current set $\{\hat{f}_i^{(n)}\}_{i=1}^{K}$ to produce a new set of estimates $\{\hat{f}_i^{(n+1)}\}_{i=1}^{K}$. The iterative approach will guarantee convergence regardless of initial choice of $f_i^{(0)}$. However, the choice of initial estimates will greatly affect the speed of convergence.

1.6 Local Optimization Schemes

Optimization is a mathematical method used to determine the minimum of an arbitrary function regardless of the function's landscape. Methods have been developed to eliminate the need to evaluate the function for every possible system dependent parameter, and instead, systematically search through the parameter space. For example, the function could be minimized over a small subset of the parameter values. Then, at the minimum of a give subset, the direction of the global minimum could be determined, which would allow for better choices for subsequent parameter sets. In this way, the entire parameter space is not explored, but the minimum can be found using a systematic approach.

A common example of optimization in molecular simulations would be the minimization of the potential energy, which could have a landscape with a high number of local minima with potentially high energy barriers between minima. In this case, it is necessary to have a systematic way to determine the global minimum without getting trapped in any local minima. Two local optimization methods are presented here, the steepest descent method and the conjugate-gradient method.

1.6.1 Steepest Descent Minimization

The Steepest Descent technique is used to find the minimum value of some function, f, with respect to some set of parameters, denoted $\overline{\lambda}$ [24]. This is done by picking a starting value, λ_0 and calculating the value of the function at that parameter value, $f(\lambda_0)$. Then calculate a direction, $\boldsymbol{s}$, in which to change $\overline{\lambda}$, given by

$$\boldsymbol{s}_0 = -\nabla f(\lambda_0). \tag{1.15}$$

Next, compute a set of n values for $\overline{\lambda}$ using

$$\lambda_n = \lambda_0 + n\alpha \boldsymbol{s_0} \tag{1.16}$$

where α is some constant step size. By doing this calculation, the set of values runs from λ_0 until the max value λ_{max} is reached, with evenly spaced steps of size α, in the direction $\boldsymbol{s_0}$. Once the simulation is run for the set of values, the function $f(\overline{\lambda})$ is minimized, and the minimum value in the set, denoted λ_{min}, becomes the new starting value.

The next run will involve calculating the gradient of the function at the previous minimum value, $\nabla f(\lambda_{\text{min}})$, in order to determine the direction to move. Then, a new set of values is generated using equation 1.16 and a simulation is run to find the minimum of the function for that set. These steps are followed until the local minimum is found, or until the minimum of a parameter set is close within a small uncertainty which is related to the statistical noise. The parameter set at the local minimum is the optimal parameter set.

1.6.2 Conjugate-Gradient Method

The Conjugate Gradient Method is formalized to solve the minimization problem for a function that can be approximated with a quadratic form [24]

$$f(\overline{\lambda}) \approx c - \boldsymbol{b} \cdot \overline{\lambda} + \frac{1}{2}\overline{\lambda} \cdot \boldsymbol{A} \cdot \overline{\lambda} \tag{1.17}$$

where $\overline{\lambda}$ is some point in N-dimensions, and both the function, $f(\overline{\lambda})$, and the gradient, $\nabla f(\overline{\lambda})$, are known or can be found.

There are two vectors that are required, denoted $\boldsymbol{g}_i$ and $\boldsymbol{h}_i$, where $i = 0, 1, 2, ...$ The first step is to let the vector $\boldsymbol{g}_0$ be arbitrary, and let $\boldsymbol{h}_0 = \boldsymbol{g}_0$. The vectors are recursively constructed as

$$\begin{aligned} \boldsymbol{g}_{i+1} &= \boldsymbol{g}_i - \lambda_i \boldsymbol{A} \cdot \boldsymbol{h}_i \\ \boldsymbol{h}_{i+1} &= \boldsymbol{g}_{i+1} + \gamma_i \boldsymbol{h}_i \end{aligned} \tag{1.18}$$

where the two vectors satisfy the “orthogonality and conjugacy conditions” [24]

$$\boldsymbol{g}_i \cdot \boldsymbol{g}_j = 0 \qquad \boldsymbol{h}_i \cdot \boldsymbol{A} \cdot \boldsymbol{h}_j = 0 \qquad \boldsymbol{g}_i \cdot \boldsymbol{h}_j = 0 \qquad j < i$$

and the scalar coefficient, γ_i, is given by the equation

$$\gamma_i = \frac{(\boldsymbol{g}_{i+1} - \boldsymbol{g}_i) \cdot \boldsymbol{g}_{i+1}}{\boldsymbol{g}_i \cdot \boldsymbol{g}_i}. \tag{1.19}$$

The formalism of the conjugate gradient method used here is to start at some point, λ_i and let the vector $\boldsymbol{g}_i$ be given by equation $\boldsymbol{g}_i = -\nabla f(\lambda_i)$. Then, let the vector $\boldsymbol{h}_i$ be the direction from λ_i that is travelled to get to the minimum along f. Next, at the minimum point along f, denoted λ_{i+1}, calculate $\boldsymbol{g}_{i+1} = -\nabla f(\lambda_{i+1})$ and then calculate the new direction to travel. The local minimum of the parameter set $\overline{\lambda}$ is determined by iteratively applying the minimization technique.

1.7 3-Letter Protein Model

The model used for the simulations is a simplified protein model presented in the paper by Bhattacherjee and Wallin [25]. In this model, all backbone atoms are represented explicitly (N,C_α,C',H,O,$H_{\alpha 1}$,$H_{\alpha 2}$) and the side chain is represented as a single larger C_β atom. The model also simplifies the amino acid types to three; polar (p), hydrophobic (h), and glycine (G). The polar and hydrophobic amino acids are represented by serine (S) and leucine (L) respectively.

For a protein with N amino acids, there are $2N$ degrees of freedom given by the dihedral angles ϕ and ψ. Other internal degrees of freedom, such as the bond lengths, are fixed to ideal values.

The energy function governing this model is a summation of 4 interaction energy terms. The energy is written as $E = E_{\text{exvol}} + E_{\text{hbond}} + E_{\text{hp}} + E_{\text{local}}$, which represent the excluded volume, hydrogen bonding, hydrophobic interaction, and the local partial charges interaction. The excluded volume energy term expands as

$$E_{\text{exvol}} = k_{\text{exvol}} \sum_{i<j} \left(\frac{\lambda_{ij} \sigma_{ij}}{r_{ij}} \right)^{12} \tag{1.20}$$

where the summation is done over all pairs, ij in the sequence. Then, r_{ij} is the distance between the pair ij, σ_{ij} is the sum of the radii of atoms σ_i and σ_j where $\sigma_i = 1.75\text{Å}, 1.55\text{Å}, 1.42\text{Å}, \text{and} 1.00\text{Å}$ for C, N, O and H atoms respectively. λ_{ij} is a scale factor, and the overall excluded-volume weight factor is $k_{\text{exvol}} = 0.1$.

The local energy term accounts for interactions between partial charges on the backbone of the protein, and is written as

$$E_{\text{local}} = k_{\text{local}} \sum_{\text{I}} \sum_{i<j} \frac{q_i q_j}{r_{ij}/\text{Å}} \tag{1.21}$$

where the summation is over all pairs of N, H, C', and O atoms in amino acid I which have partial charges of $q_i = -0.2, +0.2, +0.42$, and -0.42, respectively. The prefactor is the strength of the interaction, which is $k_{\text{local}} = 50$.

The energy due to the hydrogen bond is written as

$$\begin{aligned} E_{\text{hbond}} = k_{\text{hbond}} \sum_{ij} \gamma_{ij} \left[5\left(\frac{\sigma_{\text{hb}}}{r_{ij}}\right)^{12} - 6\left(\frac{\sigma_{\text{hb}}}{r_{ij}}\right)^{10} \right] \\ \times \left(\cos\alpha_{ij} \sin\beta_{ij} \right)^{1/2} \end{aligned} \tag{1.22}$$

where the ij summation is over all NH and CO groups, excluding nearest and next nearest neighboring groups. The prefactor in front of the summation is the hydrogen bond strength, $k_{\text{hbond}} = 3.22$, and γ_{ij} is a scale factor that depends on the types (hydrophobic, polar, or Glycine) of amino acids involved in the pair. $\gamma_{ij} = 1.0$ for hh, hp, and pp hydrogen bonds, and $\gamma_{ij} = 0.75$ for GG, Gh, and Gp pairs. The Leonard-Jones like potential has $\sigma_{ij} = 2.0$Å and r_{ij} being the separation distance. The angles α_{ij} and β_{ij} are the N-H-O and H-O-C' angles, respectively.

The hydrophobic energy term is written as

$$E_{\text{hp}} = -k_{\text{hp}} \sum_{ij} e^{-(r_{ij}-\sigma_{\text{hp}})^2/2} \tag{1.23}$$

where the sum is over all hydrophobic C$_\beta$ atoms, except nearest and next-nearest neighbors. The exponential depends on the difference between the separation distance, r_{ij} of the pair, and the optimal distance for a hydrophobic contact, which is given as $\sigma_{ij} = 5.0\text{Å}$. The strength of the hydrophobic interaction is $k_{\text{hp}} = 0.805$.

Chapter 2

Systematic Coarse-Graining Method for Optimizing Model Parameters

The objective of this research was to develop a novel systematic method to determine optimal model parameters for coarse-grained molecular simulations. The method is based on minimizing the relative entropy between a target ensemble and an ensemble that was generated from a coarse-grained model. In other words, the objective was to minimize the relative entropy, $S_{\text{rel}}(\overline{\lambda})$, with respect to a set of CG model parameters, $\overline{\lambda}$, to obtain the optimal parameter set, $\overline{\lambda}_{\text{opt}}$. Here the set of unknown model parameters is denoted as a vector $\overline{\lambda} = (\lambda_1, \ldots, \lambda_{\text{K}})$, where λ_i is the i^{th} model parameter, and K is the total number of unknown model parameters.

The equation for the relative entropy in the canonical ensemble is given in Section 1.2 and has the form

$$S_{\text{rel}}(\overline{\lambda}) = \beta \langle U_{\text{CG}}(\overline{\lambda}) \rangle_{\text{T}} - \beta \langle U_{\text{T}} \rangle_{\text{T}} - \beta \big(F_{\text{CG}}(\overline{\lambda}) - F_{\text{T}} \big) \tag{2.1}$$

where the CG and target potential energies are U_{CG} and U_{T}, and the CG and target Helmhotlz free energies are F_{CG} and F_{T}.

The concept of using the relative entropy for CG modeling was proposed by M. S. Shell, and later applied to a CG model in an article by Charmichael and Shell [11, 26]. The main challenge in determining the absolute value of the relative entropy, $S_{\mathrm{rel}}(\overline{\lambda})$, is the calculation of the free energies, $F_{\mathrm{CG}}(\overline{\lambda})$ and $F_{\mathrm{T}}(\overline{\lambda})$. There are many evaluation schemes for calculating the free energy of a molecular simulation, however, they are all computationally intensive [27]. For this reason, the approach by Charmichael and Shell was based on calculating gradients, $\partial S_{\mathrm{rel}}/\partial \lambda_i$, in order to avoid calculating F_{CG} and F_{T}. However, gradient-based methods will be local in nature and there is a benefit in determining the absolute value of the relative entropies.

2.1 Multiparameter Simulation Method

Here, we developed a novel approach for calculating the Helmholtz free energies, $F_{\mathrm{CG}}(\overline{\lambda})$, called the multiparameter method. The multiparameter method is a generalized ensemble approach, similar to the simulated tempering method in Section 1.4. In simulated tempering, the temperature, T, was the control parameter that was allowed to vary during the simulation. In contrast, in the multiparameter method, the control parameters are chosen to be a set of CG model parameters, $\overline{\lambda}$. The multiparameter simulation is carried out over a pre-selected set of parameter values that can be sampled, $\overline{\lambda}_1, \ldots, \overline{\lambda}_K$.

The probability of being in configuration, $\overline{r}$, with control parameter set, $\overline{\lambda}$ is given by the joint probability distribution,

$$p(\overline{r}, \overline{\lambda}) \propto e^{-\beta U(\overline{r}, \overline{\lambda}) + h(\overline{\lambda})}. \tag{2.2}$$

The control parameter dependent function, $h(\overline{\lambda})$, is analogous to the $g(\beta)$ function in simulated tempering techniques described in Section 1.4.

2.1.1 Free Energy

The marginal distribution of the control parameters, $\overline{\lambda}$, is obtained from the joint probability distribution by summing over all configurations, $\overline{r}$. The marginal distribution is then

$$p(\overline{\lambda}) \propto \sum_{r} p(\overline{r},\overline{\lambda}) \propto e^{-\beta F(\overline{\lambda})+h(\overline{\lambda})} \tag{2.3}$$

where the free energy of the system is $F(\overline{\lambda}) = -1/\beta \log \sum_r e^{-\beta U(\overline{r},\overline{\lambda})}$.

The marginal distribution will be flat when $h(\overline{\lambda}) = \beta F(\overline{\lambda})$. The free energy is obtained directly from the multiparameter method during a simulation by making an initial guess for $h(\overline{\lambda})$, and "tuning" it until it gives a marginal distribution that is roughly flat. The tuning process will directly give the optimal target function, $\widetilde{h}(\overline{\lambda}) = \beta F(\overline{\lambda})$ for a flat distribution, or a very good estimate, $\widetilde{h}(\overline{\lambda}) \sim \beta F(\overline{\lambda})$, if the marginal distribution is only roughly flat.

Practically, the tuning process takes a target probability distribution, $\widetilde{p}(\overline{\lambda})$, which depends on the target function, $\widetilde{h}(\overline{\lambda})$. The target distribution is defined as

$$\widetilde{p}(\overline{\lambda}) \propto e^{-\beta F(\overline{\lambda})+\widetilde{h}(\overline{\lambda})} \tag{2.4}$$

and the ratio between the two distributions is

$$\frac{\widetilde{p}(\overline{\lambda})}{p(\overline{\lambda})} \propto \frac{e^{-\beta F(\overline{\lambda})+\widetilde{h}(\overline{\lambda})}}{e^{-\beta F(\overline{\lambda})+h(\overline{\lambda})}} \propto e^{\widetilde{h}(\overline{\lambda})-h(\overline{\lambda})}. \tag{2.5}$$

Then, for $\widetilde{p}(\overline{\lambda}) = \text{constant}$, the target control parameter, $\widetilde{h}(\overline{\lambda})$ is found by rearranging the above equation to give

$$\widetilde{h}(\overline{\lambda}) = h(\overline{\lambda}) - \ln p(\overline{\lambda}) + \text{constant}. \tag{2.6}$$

Using the choice that $\widetilde{h}(\overline{\lambda}) = \beta F(\overline{\lambda})$, we can obtain the CG Helmholtz free energy by,

$$\beta F(\overline{\lambda}) = h(\overline{\lambda}) - \ln p(\overline{\lambda}) + \text{constant}. \tag{2.7}$$

2.1.2 Acceptance Criteria

The multiparamter simulation method must have two different types of Metropolis Monte Carlo updates: an update of the configuration ($\overline{r} \rightarrow \overline{r}'$), or an update of the control parameter value ($\overline{\lambda} \rightarrow \overline{\lambda}'$). Both updates use the joint probability distribution from Eq. 2.2, and have a general acceptance probability with the form

$$P_{\text{acc}}(a \rightarrow b) = \min\Bigg(1, \frac{p(\overline{r}', \overline{\lambda}')}{p(\overline{r}, \overline{\lambda})}\Bigg). \tag{2.8}$$

The update in the configuration, ($\overline{r} \rightarrow \overline{r}'$), has an acceptance probability

$$P_{\text{acc}}(\overline{r} \rightarrow \overline{r}', \overline{\lambda}) = \min\Big(1, \exp\big[-\beta \Delta U\big]\Big) \tag{2.9}$$

where the $h(\overline{\lambda})$ functions cancel out, and the change in energy, $\Delta U = U(\overline{r}', \overline{\lambda}) - U(\overline{r}, \overline{\lambda})$. This form is identical to the Metropolis acceptance criteria described in Section 1.3.

The update in the control parameter, ($\overline{\lambda} \rightarrow \overline{\lambda}'$), has an acceptance probability of

$$P_{\text{acc}}(\overline{r}, \overline{\lambda} \rightarrow \overline{\lambda}') = \min\Big(1, \exp\big[-\beta\big(U(\overline{r}, \overline{\lambda}') - U(\overline{r}, \overline{\lambda})\big) + h(\overline{\lambda}') - h(\overline{\lambda})\big]\Big). \tag{2.10}$$

The multiparameter method described here is a key part of the novel systematic CG method, as it provides a simple method for determining the Helmholtz free energy required for the relative entropy calculation.

2.2 The Steps of the Systematic CG Method

The novel systematic CG method to determine the optimal CG model parameters relies on minimizing the relative entropy, S_{rel}. The main challenge of calculating the relative entropy was determining the Helmholtz free energy, however, this challenge is overcome by using the novel multiparameter method described above. Therefore, the relative entropy in the canonical ensemble can now be calculated for a CG ensemble with a set of unknown model parameters.

The optimal CG model parameter set, $\overline{\lambda}_{\text{opt}}$, is determined by coupling the relative entropy calculation with an iterative line minimization technique to efficiently and systematically search through parameter space. The full method for optimizing the CG model parameters can be broken into multiple steps, which are listed below.

Find the optimal parameter set; iterative process

1. Choose an initial parameter set, $\overline{\lambda}$, and initial direction, $\overline{g}$.

2. Discretize each parameter, $\overline{\lambda}_j$, in the set, $\overline{\lambda}$, into N discrete values along a line in parameter space. The discrete parameters are generated according to

$$\lambda_i = \lambda_0 + i\alpha\overline{g} \tag{2.11}$$

for $i = 0, 1, \ldots, \mathrm{N} - 1$, and step size α.

Note: The number of discrete values, N, is arbitrary, but the range must remain small enough such that the energy difference between two adjacent parameter sets is small. This eliminates the issue of needing to overcome large energy barriers when moving the system from one discrete parameter set to another. This allows the system to be able to sample all states, regardless on energy barriers.

3. Run a multiparameter simulation with $\overline{\lambda}$ as the control parameter. The Monte Carlo update in the control parameter value allows it to go from $(\overline{\lambda}_i \rightarrow \overline{\lambda}_{i-1})$ or $(\overline{\lambda}_i \rightarrow \overline{\lambda}_{i+1})$.

   From the simulation, obtain the CG Helmholtz free energy, $F_{\text{CG}}(\overline{\lambda})$, as described in Sect. 2.1.1, as well as the ensemble average of the CG energy, $\langle U_{\text{CG}}(\overline{\lambda})\rangle$.

4. Apply the self-consistent estimate equation (MBAR method) to obtain an improved estimate of the CG free energy, $F_{\text{CG}}(\overline{\lambda})$.

5. Calculate the CG energy in the target ensemble, $\langle U_{\text{CG}}(\overline{\lambda})\rangle_{\text{T}}$, for the current parameter set $\overline{\lambda}$.

6. Calculate the $\overline{\lambda}$-dependent part of the canonical ensemble relative entropy using information from steps 3 to 5,

$$S_{\text{rel}}(\overline{\lambda}) = \beta\langle U_{\text{CG}}(\overline{\lambda})\rangle_{\text{T}} - \beta F_{\text{CG}}(\overline{\lambda})$$

7. Find the index of $\overline{\lambda}_j$ for which the relative entropy is the lowest. Let this be the new initial parameter set, $\overline{\lambda}$.

8. Calculate the gradient of the relative entropy for the new parameter set, $\nabla S_{\text{rel}}(\overline{\lambda})$.

9. Pick the new direction following the steepest-descent or conjugate gradient method.

   Steepest-Descent: Let the new direction be, $\overline{g} = -\nabla S_{\mathrm{rel}}(\overline{\lambda})$.

   Conjugate gradient: Let the new direction be $\overline{g} = \overline{h}$, which is given in Eq. 1.18.

10. Repeat steps 2 through 9 until the relative entropy reaches the local minimum (when $|\nabla S_{\mathrm{rel}}| < \epsilon$). Let the parameter set at the local minimum be the optimal CG model parameter set, $\overline{\lambda}_{\mathrm{opt}}$.

It is important to note that each iteration of the systematic CG method requires a new multiparamter simulation to be run, with a new set of model parameters, $\overline{\lambda}$.

2.3 Validation of the Systematic CG Method

Two tests were done to prove the validity of the systematic CG method. Both tests were done using the 3-letter CG protein folding model described in Section 1.7. The target ensemble was defined as the ensemble generated by a Metropolis MC simulation for an equilibrium set of model parameters. For the first validation test, the model ensemble was generated from the multiparameter simulation when one of the known model parameters from the CG model was discretized. Then, by calculating the canonical ensemble definition of the relative entropy between the model ensemble and the target ensemble, and using the line minimization technique, the multiparameter method was used to recapture the target value.

The second validation test was to extend the above definition of the model ensemble to be the ensemble generated by the multiparameter simulation when two known model parameters were discretized. Similar to the first test, the relative entropy and line minimization techniques were used to recapture the target set of parameters simultaneously.

2.3.1 Recapturing One Known Parameter

As a first test, we apply our systematic CG method to the 3-letter CG model described in section 1.7 with a single free model parameter, $\overline{\lambda} = k_{\mathrm{hp}}$, where k_{hp} is the strength of the hydrophobic interaction given in Eq. 1.23.

The model potential energy function can be simplified as

$$E(r, \lambda_j) = E_0(r) + \lambda_j e_{\mathrm{hp}}(r) \tag{2.12}$$

where the total energy depends on the configuration, r, and the value and index of the dynamic parameter, λ_j. Here, E_0 is the sum of the energy terms excluding the hydrophobic energy, and $e_{\mathrm{hp}} = -\sum_{ij} e^{-(r_{ij}-\sigma_{\mathrm{hp}})^2/2}$ is the part of the hydrophobic energy that does not depend on k_{hp}.

The multiparameter simulation method was used here, and the single model parameter was discretized into a set of N parameters, where the correct (target) value of of the hydrophobic interaction strength, $\lambda_{\mathrm{T}} = k_{\mathrm{hp}} = 0.805$, was one of the N discrete values. Mathematically, the discrete parameter set is $\overline{\lambda} = (\lambda_1, \ldots, \lambda_{\mathrm{T}}, \ldots, \lambda_N)$, where $N = 10$ here.

The relative entropy, given by Eq. 1.6, was simplified significantly using the energy function given in Eq. 2.12. The first term of the relative entropy simplifies as

$$\begin{aligned}\langle U_{\mathrm{CG}} - U_{\mathrm{T}} \rangle_{\mathrm{T}} &= \langle (E_0 + \lambda_j e_{\mathrm{hp}} - E_0 - \lambda_{\mathrm{T}} e_{\mathrm{hp}}) \rangle_{\mathrm{T}} \\ &= (\lambda_j - \lambda_{\mathrm{T}}) \langle e_{\mathrm{hp}} \rangle_{\mathrm{T}} \\ &= \frac{(\lambda_j - \lambda_{\mathrm{T}})}{\lambda_{\mathrm{T}}} \langle E_{\mathrm{hp}} \rangle_{\mathrm{T}}\end{aligned} \tag{2.13}$$

where λ_j is the j-th dynamic parameter value in the model system, and λ_T is the target value. $\langle E_{\mathrm{hp}} \rangle_{\mathrm{T}}$ is the ensemble average of the CG hydrophobic energy term calculated in the target ensemble. The second term of the relative entropy is $\beta(F_{\mathrm{CG}} - F_{\mathrm{T}})$, where the simulation calculates the free energy $h_{\mathrm{CG}} = \beta F_{\mathrm{CG}}$. Lastly, since the relative entropy was being minimized, the exact value was not required to be known, and thus, the constant mapping term was ignored.

Therefore, the simplified relative entropy is given as

$$S_{\mathrm{rel}}(\lambda_j) = \frac{(\lambda_j - \lambda_{\mathrm{T}})}{\lambda_{\mathrm{T}}} \langle E_{\mathrm{hp}} \rangle_{\mathrm{T}} - (h_j - h_{\mathrm{T}}). \tag{2.14}$$

The relative entropy is minimized with respect to a target ensemble, which can first be defined as all of the configurations for which the index, j, corresponds to the case when $\lambda_j = k_{\mathrm{hp}} = 0.805$, which is the target value. The simplified relative entropy (Eq. 2.14) will be zero, $S_{\mathrm{rel}}(\lambda_{\mathrm{T}}) = 0$, by definition when the dynamic parameter equals the target value, $\lambda_j = \lambda_{\mathrm{T}}$.

To determine $S_{\mathrm{rel}}(\lambda_j)$ for $j = 1, \ldots, 10$, we carried out 20 multiparameter simulations each with 20 million MC cycles, which had a runtime of around 20 hours. The disctretized set of parameters for the hydrophobic strength were chosen to be in the range $\lambda = [0.78, 0.825]$. The results for the relative entropies, $S_{\mathrm{rel}}(\overline{\lambda})$, calculated for the 20 simulations are given in Fig. 2.1.

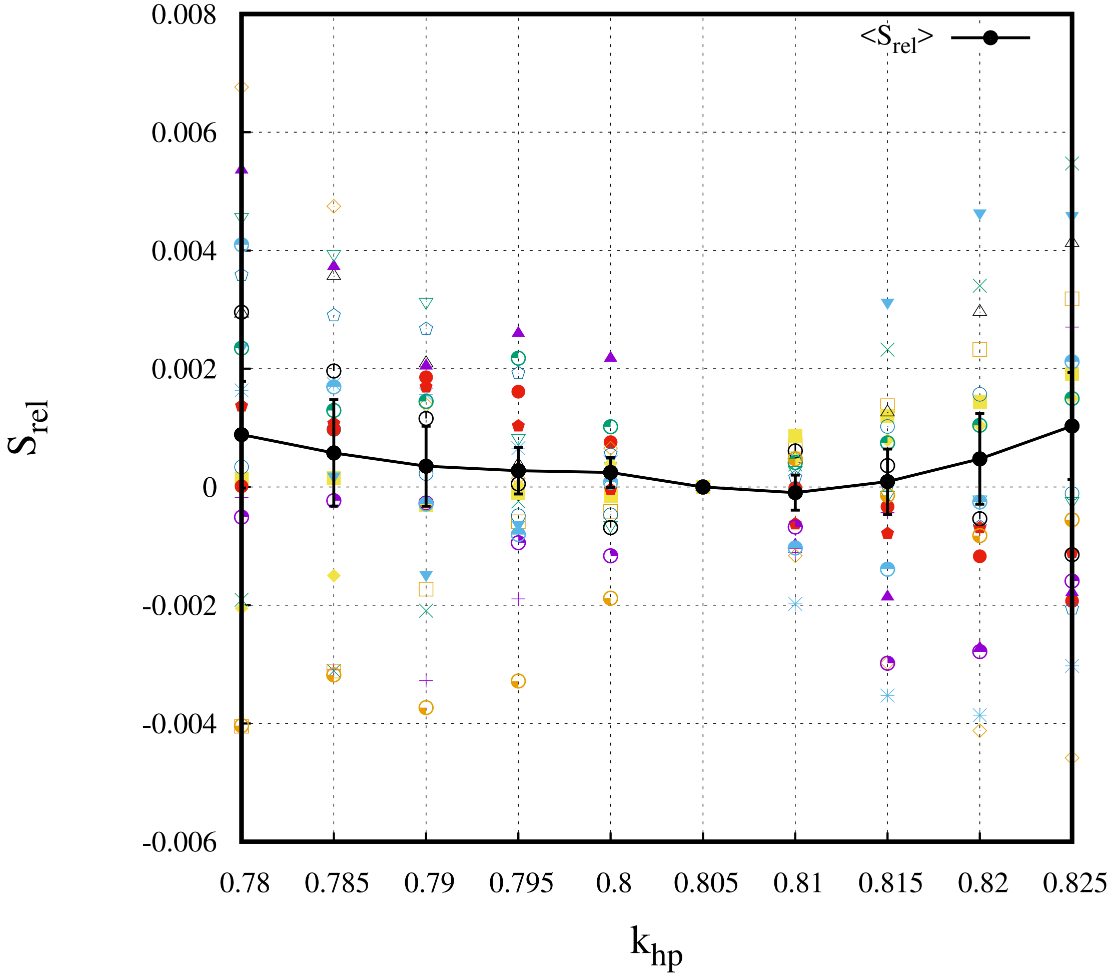

Figure 2.1: Average Relative Entropy, S_{rel} of 20 multiparameter simulations plotted versus the strength of the hydrophobic interaction, k_{hp} (solid line), and the individual relative entropy from each simulation (symbols). The standard error for the average relative entropy is given by the error bars.

As expected, the relative entropy is zero when the dynamic parameter equals the target value, however, the statistical errors are large here. There is a minimum at $\lambda = 0.81$ instead of at $\lambda = 0.805$, and the range of the relative entropy at each index is quite large, with a significant number of data points being negative.

The large spread in $S_{\mathrm{rel}}(\lambda_j)$ values in Fig. 2.1 is due to the large statistical errors present in the values of the free energies, $F_{\mathrm{CG}}(\lambda_j)$. In order to improve the free energies, $F_{\mathrm{CG}}(\overline{\lambda})$, we apply the MBAR estimate equation, described in section 1.5. This method is a quick calculation that uses information about the ensemble of generated configurations from the simulation to obtain a better estimate for the free energies. The MBAR calculation significantly improves the results, as shown in Fig. 2.2, which is identical to the previous graph, except the relative entropy is calculated with the corrected free energy.

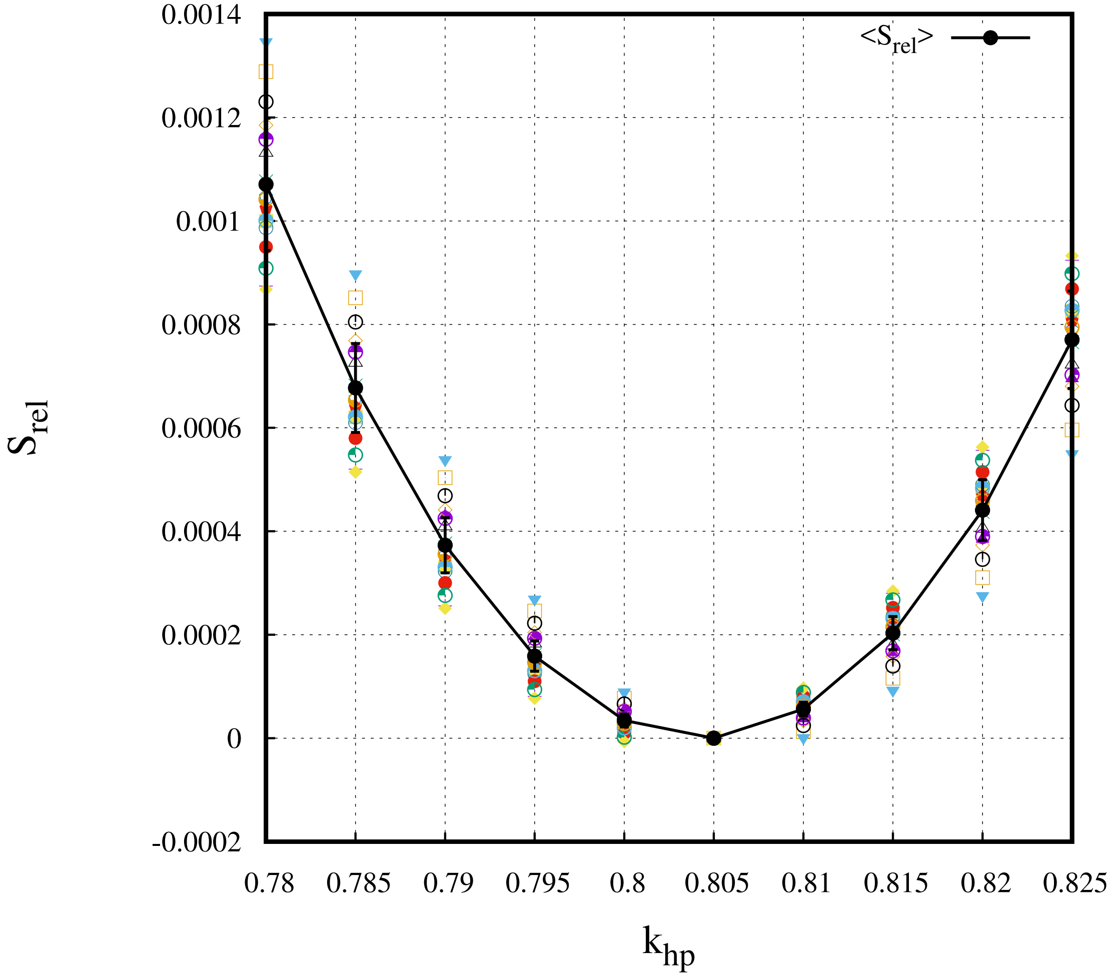

Figure 2.2: Average Relative Entropy, S_{rel}, of 20 multiparameter simulations, with MBAR estimate of free energy, plotted versus the strength of the hydrophobic interaction, k_{hp} (solid line), and the individual S_{rel} from each simulation (symbols). The standard error for the average relative entropy is given by the error bars.

It is quite clear that the self-consistent free energy correction greatly improves the results of the relative entropy calculation. Again, by definition, the relative entropy is zero at the target value, but now the entropy everywhere else is greater than zero. This is the expected result, because there should be a minimum in the entropy when the dynamic parameter equals the target parameter value.

The above test of the systematic CG method is flawed slightly in that the target ensemble used to minimize the relative entropy was generated from the multiparameter simulations. A more robust test of the method would involve a target ensemble obtained from a separate simulation. This was done here, and the target energy, $\langle E_{\mathrm{CG}}(\overline{\lambda})\rangle_{\mathrm{T}}$, was obtained by running 10 independent simulations where the value of the hydrophobic strength was fixed at $k_{\mathrm{hp}} = 0.805$. Each simulation was run for 20 million MC cycles, and the average hydrophobic energy was calculated over all configurations and found to be $\langle E_{\mathrm{hp}}\rangle_{\mathrm{T}} = -2.880781 \pm 0.0007639$. The target free energy is not required for the relative entropy calculation since it will only shift the values of the relative entropy by a constant, and not changing the location of the minimum. However, the self-consistent MBAR free energy calculation is done to improve the CG model free energy function. The average relative entropy with independent target is plotted in the same way as before, given in Fig. 2.3.

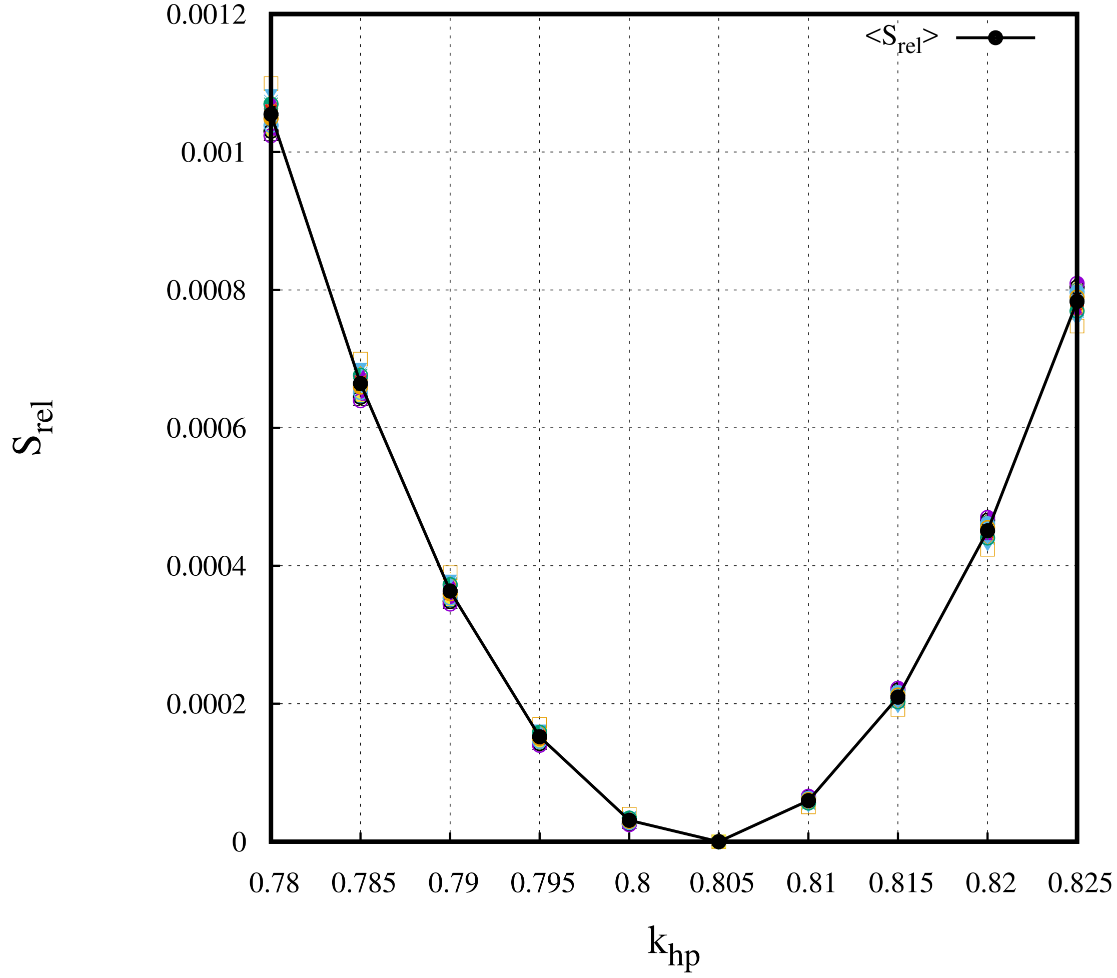

Figure 2.3: Average Relative Entropy, S_{rel}, of 20 multiparameter simulations, with the MBAR free energy estimate, plotted versus the strength of the hydrophobic interaction, k_{hp} (solid line), and the individual S_{rel} from each simulation (symbols). The standard error for the average relative entropy is given by the error bars. The target ensemble here is multiple independent fixed parameter simulation

The relative entropy is even better when the target ensemble is an independent simulation. The standard error is included on the graph, but cannot be seen because they are extremely small. This shows that the MBAR free energy calculation and an independent target ensemble both have a significant impact on the quality of the relative entropy statistics.

2.3.2 Minimization using Steepest Descent

In order to test the calculation of the gradients, $\partial S_{\text{rel}}/\partial\lambda$, and to determine the precision with which the an unknown model parameter, λ, can be found, we tested a minimization scheme on the above system. Carmichael and Shell [26] proposed using either the Newton-Raphson or steepest descent minimization technique to minimize the relative entropy. These techniques were used to iteratively step through parameter space in the direction of the global minimum, such that each iteration brought the CG model parameters closer to the optimum. The steepest descent technique can be written

$$\lambda^{k+1} = \lambda^k - \alpha\frac{\partial S_{\text{rel}}}{\partial\lambda} \tag{2.15}$$

for some step-size, α. The derivative of the relative entropy is calculated using Eq. B.7 for the reweighted gradient, which is found in Appendix B. The iterative process is valid if the current parameter, λ, is close to the initial guess parameter, λ^0. This is the case if the initial guess parameter is chosen to be close to the target $\lambda = 0.805$. Thus, choosing $\lambda^0 = 0.80$ or even $\lambda^0 = 0.78$ will satisfy the validity condition. Using the configurations saved from the multiple simulations, the derivative can be calculated and the minimum can be found using this approach.

The iterative process is repeated until the current parameter set and the next parameter set has a difference that is less than $\epsilon = 1.0 \times 10^{-8}$. The initial guess parameter was chosen to be $\lambda^0 = 0.80$, and the step size was varied. The results are shown in Table 2.1.

The number of iterations rapidly increased as the step size was decreased, but the results for the minimum parameter did get closer to the target of $\lambda = 0.805$ as the step size was decreased.

From the minimization of the relative entropy and the minimization of the en-

α	$\langle\lambda_{\text{min}}\rangle$	$\langle N_{\text{iterations}}\rangle$
0.05	0.80567151	56.8
0.01	0.80567130	264.15
0.001	0.80566886	2050.4
0.0001	0.80564449	14292.6

Table 2.1: Steepest Descent iterative approach to determine the minimum parameter, λ_{min}, as a function of step size, α.

tire parameter space using the derivative of the relative entropy, it is clear that the target value of $\lambda = 0.805$ was very successfully obtained with a very small degree of uncertainty. The uncertainty here is due to statistical noise, and by using the MBAR calculation and the independent target ensemble, the noise was significantly decreased.

2.3.3 Recapturing Two Known Parameters

The second validation test was an extension of the first test, and showed that the method can optimize two model parameters simultaneously. The strength of the hydrophobic interaction remained dynamic, and the positive coefficient in the hydrogen bond energy term, of Eq. 1.22, was also discretized. This coefficient is the strength of the hydrogen bond interaction, and is known to be $k_{\text{hbond}} = 3.22$. One important difference between this test and the previous test is that the dynamic parameter, λ, is now a vector with two components, $\overline{\lambda} = (\lambda_1, \lambda_2)$ where $\lambda_1 = k_{\text{hp}}$ and $\lambda_2 = k_{\text{hb}}$. Each of the components were discretized into 10 values as, $\lambda_1 = (\lambda_1^{(1)}, \ldots, \lambda_1^{(10)})$ and $\lambda_2 = (\lambda_2^{(1)}, \ldots, \lambda_2^{(10)})$. Then, during the multiparameter simulation, there was an equal chance that either one of the components would undergo an update; $(\lambda_1 \rightarrow \lambda_1')$ or $(\lambda_2 \rightarrow \lambda_2')$. In this way, the multiparameter simulation was able to sample the set of 100 unique pairs of the two parameters.

For a multiparameter simulation of two dynamic parameters, the number of MC cycles must be increased to ensure sufficient sampling at every point in parameter space. There were 10 simulations run, each with 40 million MC cycles to ensure good sampling. The relative entropy was calculated using the free energy calculated during the simulation, and using the MBAR calculation. Both relative entropy calculations used the definition for the "target" ensemble to be the same independent set of simulations as before. Thus, the target hydrogen bond interaction energy term was $\langle E_{\text{hb}} \rangle_{\text{T}} = -22.575294 \pm 0.0124634$, and the target hydrophobic energy was the same as before.

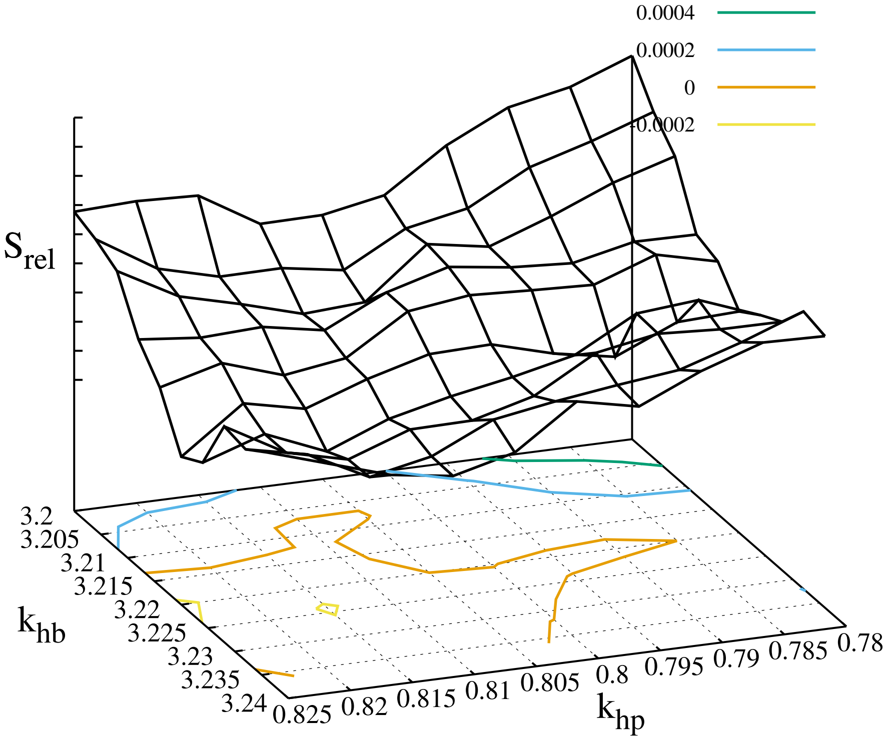

Figure 2.4: Average Relative Entropy of 10 multiparameter simulations plotted versus the strength of the hydrophobic interaction and the strength of the hydrogen bond. The relative entropy was calculated using the free energy obtained from the simulation.

Fig. 2.4 shows the average relative entropy at each point in the 2-dimensional parameter space, connected together with a mesh to show the relative entropy landscape. The landscape is very noisy, with a minimum at $(\lambda_{\text{hp}}, \lambda_{\text{hb}}) = (0.815, 3.225)$. It is important to note that the relative entropy scale is very small, and thus, any amount of statistical noise would significantly affect the landscape. This is another reason why the self-consistent free energy calculation to get a better estimate is important.

Fig. 2.5 shows the average relative entropy landscape when the entropy is calculated using the MBAR estimate for free energy.

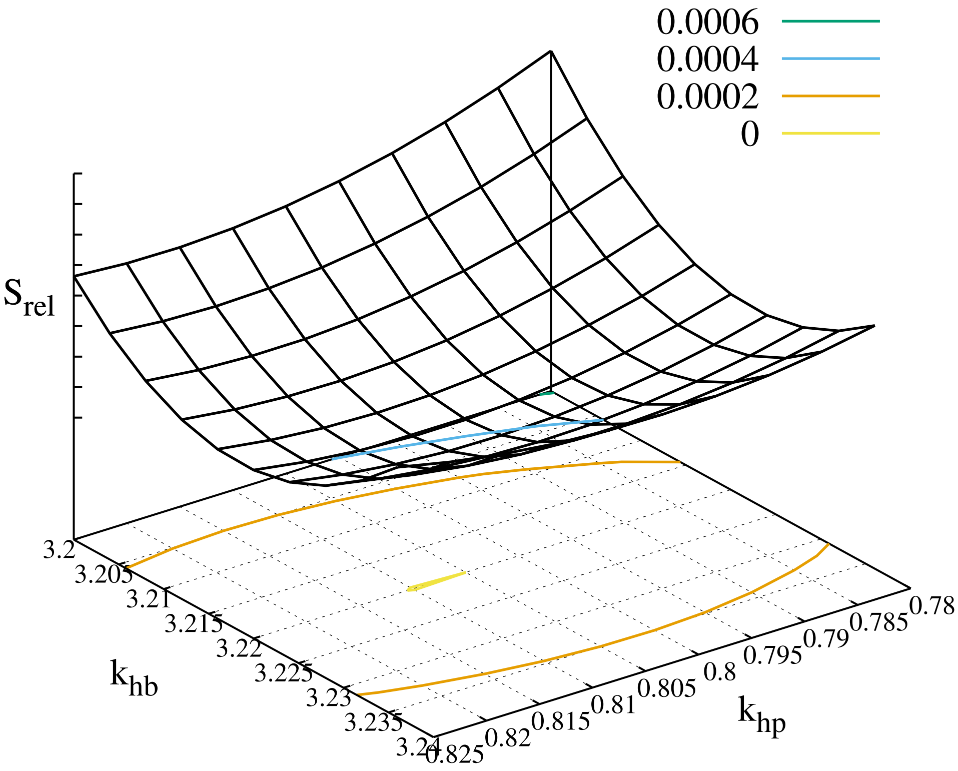

Figure 2.5: Average Relative Entropy of 10 multiparameter simulations plotted versus the strength of the hydrophobic interaction and the strength of the hydrogen bond. The relative entropy was calculated using the self-consistent free energy estimator equation.

From Fig. 2.5, it is clear that the MBAR estimated free energies greatly improved the entropy landscape. The relative entropy now shows a smooth landscape with no local minima, and a very clear global minimum that occurs at $(\lambda_{\text{hp}}, \lambda_{\text{hb}}) \approx (0.805, 3.22)$, which is equal to the target values for the strength of the hydrophobic and hydrogen bond interactions. The fact that the global minimum has two values for λ_{hp} is due to the fact that by definition, the relative entropy at the target dynamic parameter values is exactly zero. However, the relative entropy at $\lambda_{\mathrm{hp}} = 0.81$ is not exactly zero, but equal to zero within the uncertainty.

Although the above results were successful in recapturing the known values for one or two CG model parameters, these methods do not scale well with increased number of parameters. For example, if 3 parameters were discretized, the dynamic parameter λ would be a 3-dimensional matrix with 1000 points. The number of MC cycles required to have good sampling at each point would be extremely large, and any more than 3 dynamic parameters would be impossible to study.

2.3.4 Testing Line Minimization Schemes

Here, we couple the multiparameter simulation method with line minimization techniques. Local minimization methods are ideal for this problem for many reasons; the local minimum can be found in a systematic and direct way while only using a small subset of the parameter space. Therefore, coupling the method with line minimization schemes will allow higher dimensional parameter space to be searched through efficiently and without a significantly higher computational cost.

2.3.4.1 Line Minimization with Steepest-Descent Method

The first minimization method used was the Steepest-Descent method, in which the parameter λ is updated according to the equation

$$\lambda^{k+1} = \lambda^{k} - \alpha \frac{\partial S_{\mathrm{rel}}}{\partial \lambda} \tag{2.16}$$

for some step-size, α.

To apply the steepest-descent method here, a starting point in parameter space is chosen, $(\lambda^0_{\mathrm{hp}}, \lambda^0_{\mathrm{hb}})$, and an arbitrary step size, and an arbitrary initial direction.

The initial parameter set is then generated using equation 1.16. A simulation is then run for the given parameter set, and the relative entropy is calculated using the self-consistent free energy estimate. The minimum of the relative entropy is found and the gradient is calculated using Eq. B.7 from Appendix B. The next parameter set starts at the relative entropy minimum, and goes in the direction of the negative gradient. This process is continued until the parameters at the relative entropy minimum are equal to the target values, within a small finite difference. Results for 4 simulations are shown in Fig. 2.6, where the target global minimum is symbolized with a square and denoted λ_{target}, while the minimum obtained from the line minimization method is denoted λ_{opt}.

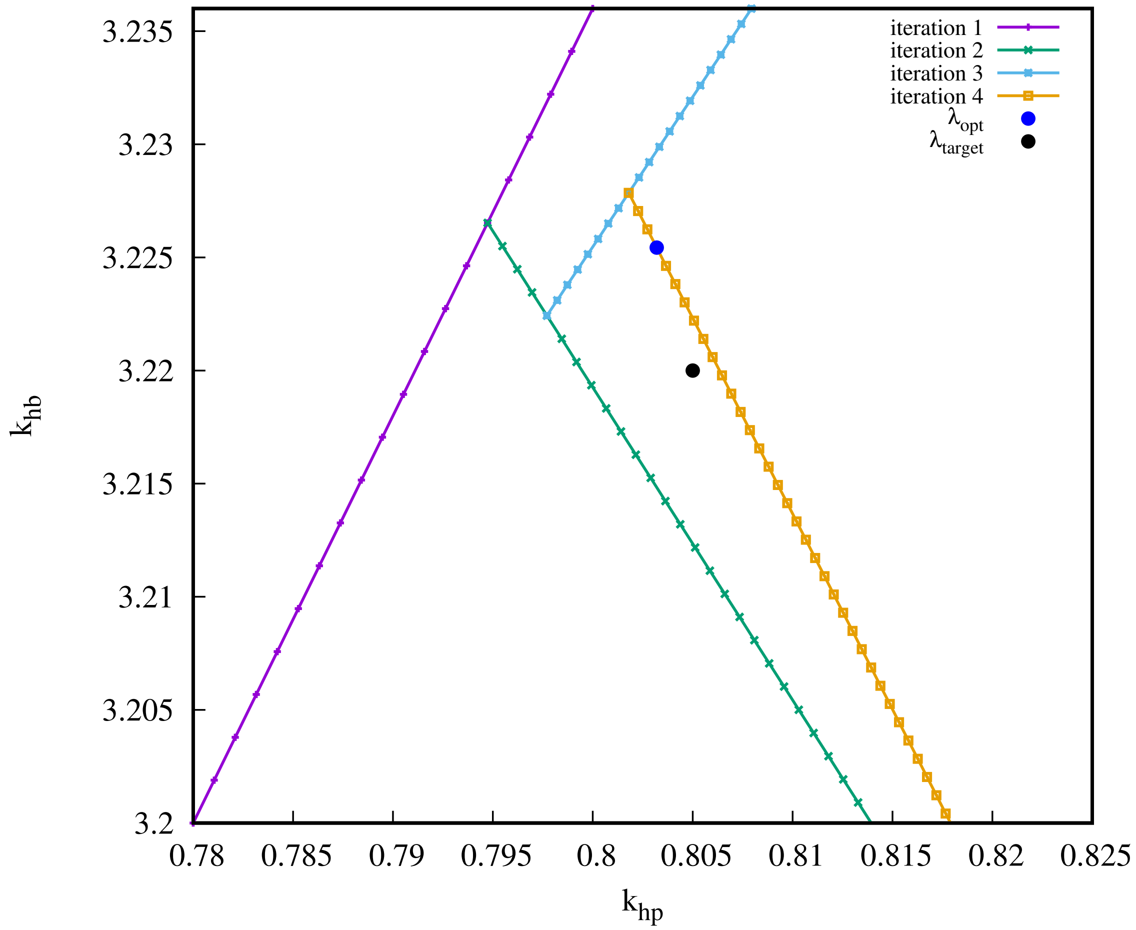

Figure 2.6: Results for the 4 iterations required to find the optimal model parameter set using the systematic CG method and the steepest-descent minimization method.

Unfortunately, the steepest-descent method was incapable of accurately recapturing the target global minimum of $(\widetilde{\lambda}_{\mathrm{hp}}, \widetilde{\lambda}_{\mathrm{hb}}) = (0.805, 3.22)$. The minimum that the method recovered was $(\lambda_{\mathrm{hp}}, \lambda_{\mathrm{hb}}) = (0.803666, 3.224626)$. Although the difference between the target global minimum and the achieved global minimum looks large in the figure, the difference is $(\Delta\lambda_{\mathrm{hp}}, \Delta\lambda_{\mathrm{hb}}) = (-0.001334, 0.004626)$. The method was stopped at this point due to the very small difference between the relative entropy for the 4th simulation, and any additional simulations would be greatly affected by statistical noise.

2.3.4.2 Line Minimization with Conjugate-Gradient Method

Due to the limitations of the Steepest-Descent method, the Conjugate-Gradient method was used as a minimization scheme. The conjugate-gradient method was initialized with the parameters $(\lambda_{\mathrm{hp}}^{0}, \lambda_{\mathrm{hb}}^{0}) = (0.80, 3.20)$, and the initial gradient $g_0 = 1$. The parameter set was generated following the conjugate-gradient method described above in Section 1.6.2, and a simulation was run for the given parameter set. Results from 3 simulations are shown in Fig. 2.7.

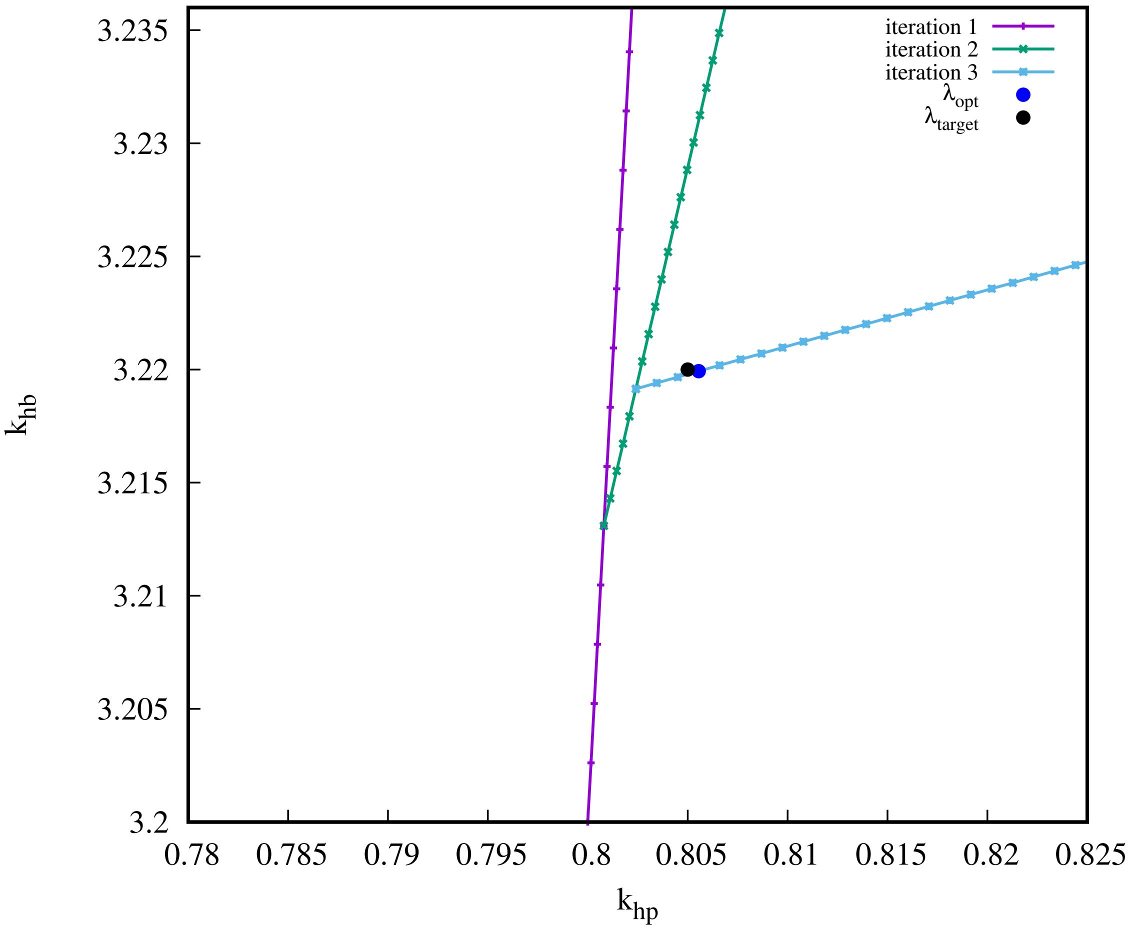

Figure 2.7: Results for the 4 iterations required to find the optimal model parameter set using the systematic CG method and the conjugate-gradient method.

The conjugate-gradient method only required 3 simulations to find the global minimum within a very small difference, $\varepsilon \sim 10^{-6}$. The final minimum was found to be $(\lambda_{\rm hp}, \lambda_{\rm hb}) = (0.805550, 3.219926)$, which gives a difference from the target global minimum of $(\Delta\lambda_{\rm hp}, \Delta\lambda_{\rm hb}) = (0.00055, -0.000074)$. It is clear from these results that the conjugate-gradient method is very efficient at recapturing the global minimum of a multiparameter simulation with 2 dynamic parameter. Very little of parameter space was explored, and the method was able to find the minimum to within an extremely small range after 3 Monte Carlo simulations.

2.4 Conclusion

It has been shown that by combining multiple methods, such as relative entropy minimization, generalized ensemble approach to simulations, MBAR free energy estimates, and local optimization, it is possible to efficiently determine the target values for CG model parameters. This systematic CG method described at the beginning of the Chapter has successfully recaptured the value of two model parameters with only 3 simulations. Therefore, the method should easily scale to allow the study of a large number of model parameters. Furthermore, the method can be applied to model parameters that are unknown, and to target systems that are not CG simulations.

Chapter 3

Application of the Systematic CG Method

To demonstrate the capability of the systematic CG method as an approach to determine the optimal coarse-grain model parameters, the method was applied to two different CG protein folding models with either 13 or 91 unknown model parameters. Two different target ensembles were considered, a single experimentally determined native structure, and a large ensemble of configurations generated from a molecular dynamics simulation.

3.1 Protein Sequence and Target Ensembles

The systematic CG method developed here relies on the minimization of the relative entropy in order to determine the optimal CG model parameter set. The relative entropy is a measure of how close the CG model ensemble is to the target ensemble, and thus, the optimal parameters are those that allow the CG model ensemble to best match the target ensemble. Therefore, the choice of the target system is crucial because it determines the optimal parameter set, $\overline{\lambda}_{\mathrm{opt}}$, as well as the extent to which the optimized CG model can be applied to other systems.

Furthermore, the choice of what protein being studied is important as well. The CG protein folding model represents all amino acid types, and allows for the protein to fold into its native configuration. Therefore, we selected the protein BBA as a test case because despite its short length of 28 residues and 504 atoms total, it contains 13 of the 20 amino acid types. This protein has a very interesting native structure consisting of two β-sheets and one α-helix, and thus contains both main types of secondary structures. It is commonly referred to as the $\beta\beta\alpha$, or BBA, protein [28].

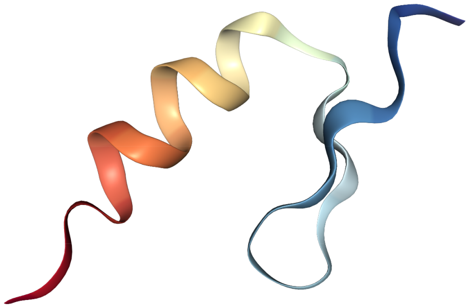

Figure 3.1: The native solution structure of the 1FME protein from an NMR experiment. The image was generated from data obtained from the RCSB Protein Data Bank [28].

The first target ensemble used for calculating the relative entropy was the single experimentally determined native configuration of the BBA protein. This target ensemble was obtained from the Protein Data Bank, with PDB id 1FME, and has the native structure from a solution NMR experiment, shown in Figure 3.1 [28].

The second target ensemble was an all-atom molecular dynamics simulation of the BBA protein done by the D. E. Shaw research group [29]. This target ensemble was not just the single native configuration, but rather a large ensemble of configurations generated over a long Molecular Dynamics (MD) simulation. The MD simulation was $223\mu s$, with configurations captured every $0.02\mu s$, resulting in $N_c = 111500$ saved configurations.

The systematic CG method described in Chapter 2 was used to obtained the optimal model parameter set by minimizing the relative entropy between the generated CG ensemble and the two different target ensembles.

3.1.1 Comparing Two Configurations

In addition to calculating the relative entropy to compare two ensembles, we also calculated the root-mean squared deviation (RMSD) to measure the structural similarity between two individual conformations. The RMSD is a measure of how close the positions of the atoms in two configurations, a and b, match. This is done by calculating the distance, $\delta_i = |\overline{r}_a^{(i)} - \overline{r}_b^{(i)}|$, where $\overline{r}_a^{(i)}$ is the position of atom i in configuration a, and $\overline{r}_b^{(i)}$ is the position of the same atom i in configuration b. This can be done for every atom in the configuration, or just some of the atoms. Here, the RMSD is calculated between the Cα atoms of a CG configuration and those of the experimentally determined structure.

The RMSD is given by the equation,

$$\text{RMSD} = \min \sqrt{\frac{1}{N} \sum_{i=1}^{N} \delta_i^2} \tag{3.1}$$

where the minimum is taken over all relative rotations and translations of the two configurations. It is clear from the equation that RMSD ≥ 0, and a low value for RMSD means the two configurations were structurally very similar.

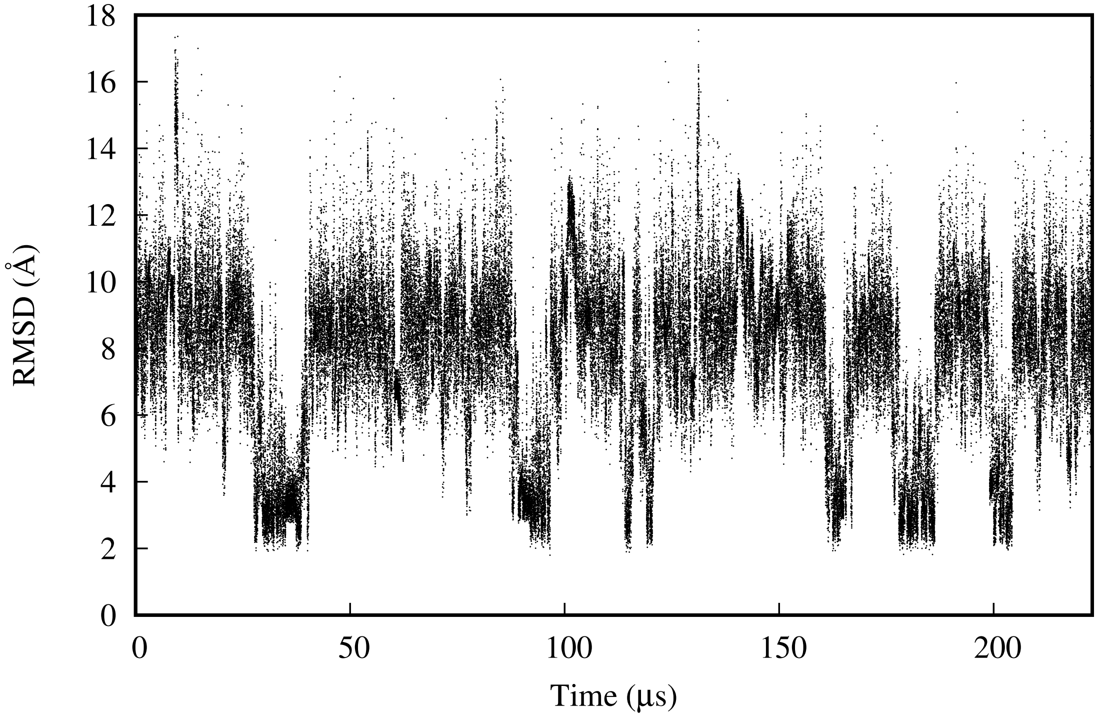

Figure 3.2: Cα-RMSD for each configuration during the 223μs all-atom Molecular Dynamics simulation from the D. E. Shaw Research group [29].

Figure 3.2 shows the RMSD as a function of time for the all-atom MD simulation of BBA from Shaw *et al.* calculated with respect to the experimentally determined structure 1FME. The lowest RMSD obtained during the MD simulation was 1.6Å [29]. This figure shows two distinct states for the protein conformation. When the RMSD is very low, around $2 - 4$Å, the protein is in its folded native state. When RMSD ≥ 5Å, the protein is in a configuration different from its native state, whether it is unfolded entirely or forming some other structure. This figure shows that over the course of the simulation, the protein folds and unfolds several times.

An alternate way to visualize the RMSD over the course of a simulation is to generate a binned histogram for the probability of having a particular value of RMSD. The histogram was generated by counting the number of times the RMSD is in the range of each bin, then plotting the probability or frequency as a function of RMSD. This type of graph is shown in Figure 3.3, using the data from the long MD simulation presented in Figure 3.2. This figure shows a bimodal shape, where the two peaks correspond to the folded and unfolded states of the BBA protein respectively.

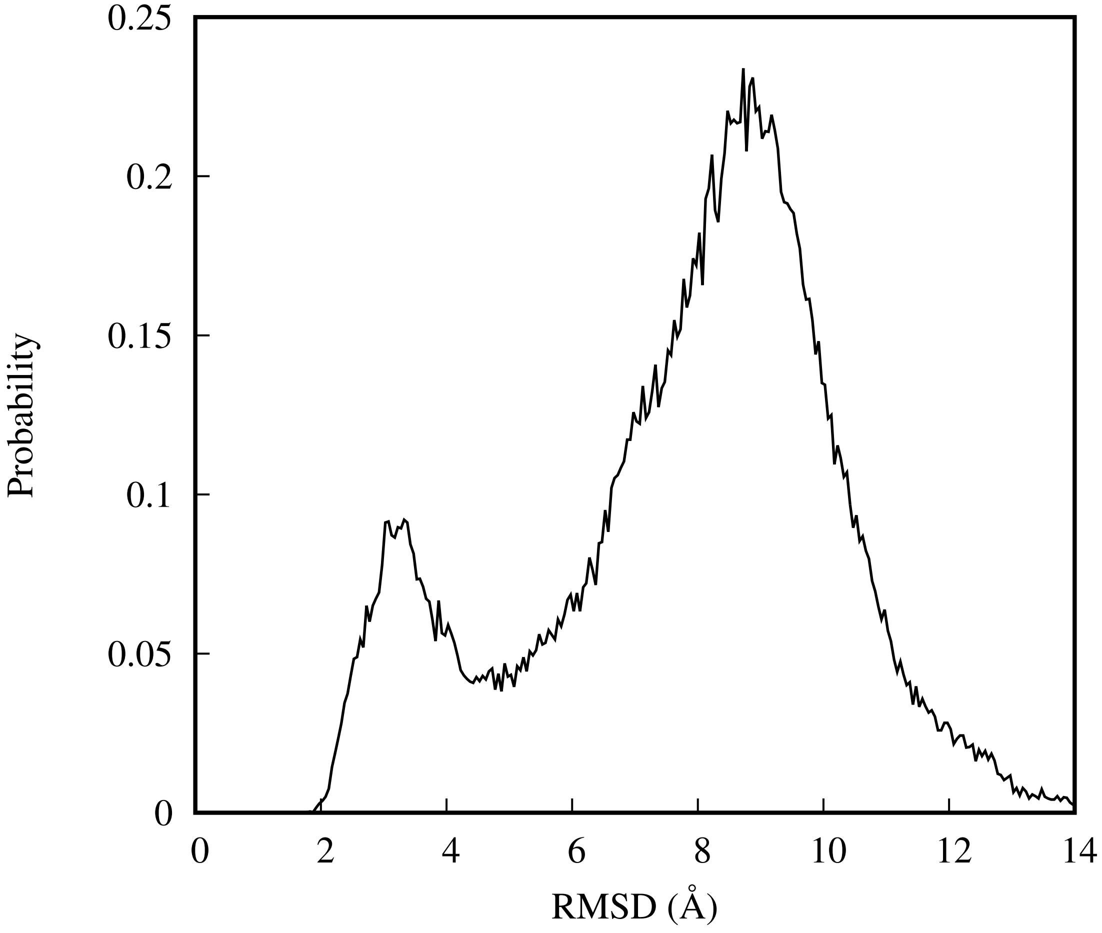

Figure 3.3: Histogram showing the probability of having a particular value of the Cα-RMSD from each configuration during the $223\mu s$ Molecular Dynamics simulation from the D. E. Shaw Research group [29]. The bin size must be small, and was chosen to be 0.05Å.

3.2 13-parameter Side-Chain Potential Energy Function

The CG protein model described in Section 1.7 had three amino acid types, polar (p), hydrophobic (h), and glycine (G). This was expanded to include all 20 amino acid types, but maintain the simplified structure of representing the side chain by a single larger Cβ atom. The hydrophobic interaction energy term was replaced with a more general side-chain interaction energy, which has the form

$$E_{sc} = -\sum_{i=1}^{N}\sum_{j=i+3}^{N} M(a_i, a_j)\varepsilon(r_{ij}) \tag{3.2}$$

where N is the total number of amino acids in the sequence, and a_i denotes the amino acid type of amino acid i, and $\varepsilon(r_{ij}) = e^{-(\sigma_{ij}-r_{ij})^2/2}$. The double summation is over all pairs of side chains, ij, ignoring nearest and next nearest neighbors along the chain.

The parameter $M(a_i, a_j)$ determines the "strength" of the interaction between a pair of amino acids with types, a_i and a_j. In principle, there are 210 parameters $M(a_i, a_j)$ determining the side-chain interactions. To decrease this number, $M(a_i, a_j)$ is defined as

$$M(a_i, a_j) = b(a_i)b(a_j) \tag{3.3}$$

where $b(a_i)$ is the side-chain interaction strength of amino acid a_i.

The exponential term is the same as the hydrophobic interaction, where r_{ij} is the separation distance between the side chain Cβ atoms at index i and j, and $\sigma_{ij} = \sigma_{\text{hp}} = 5.0$Å. This terms gives a measure of the "range" of the side chain interaction.

The side chain interaction energy as a function of configuration, r_n, can be written in a simplified form given as

$$E_{sc}(r_n) = -\sum_{i<j}^{N} b_i b_j \varepsilon(r_{ij}) \tag{3.4}$$

where the side chain strength $b_i \equiv b(a_i)$.

The unknown parameters in this energy function are the side chain interaction strengths for the amino acid types, b_i. The set of the parameters can be written as a vector with 20 components, one for each amino acid type,

$$\overline{\lambda} = (b_1, b_2, \ldots, b_{20}) \tag{3.5}$$

where the 7 amino acid types that are not present in the BBA protein all have interaction strengths of zero.

The systematic CG method defined in Chapter 2 will be used to determine the optimal set of those side chain strength parameters by minimizing the relative entropy between the generated model ensemble and a target ensemble. The rest of this section will organize the details behind determining the optimal parameter set. The first task was simplifying and calculating the relative entropy and gradient for the new energy function. This included calculating the average of the CG energy function in the target ensembles. Then, the iterative approach was used to systematically find the optimal set. This was done separately for the two different targets, as well as for various optimization methods in order to determine the most efficient way to find the optimal parameter set.

3.2.1 Relative Entropy

The relative entropy can be simplified in a similar manner to the simplifications made in Chapter 2. Specifically, since the function is being minimized and the exact value of the relative entropy is not required, the equation can be simplified by lumping all of the constant or parameter independent terms together.

The relative entropy is again given by the equation

$$S_{\text{rel}} = \beta \langle U_{\text{CG}} - U_{\text{T}} \rangle_{\text{T}} - \beta \big(F_{\text{CG}} - F_{\text{T}} \big) + \langle S_{\text{map}} \rangle_{\text{T}}. \tag{3.6}$$

Here, the mapping entropy and the target ensemble free energy, F_{T} are both independent of the parameter set, and thus will be absorbed into the constant, denoted S_{const}. The first term in Eq. 3.6 can be simplified by noting that the target ensemble potential energy is a constant over the target ensemble if the target is independent of the CG model ensemble. Therefore, $\langle U_{\text{CG}} - U_{\text{T}} \rangle_{\text{T}} \equiv \langle U_{\text{CG}} \rangle_{\text{T}} + C$, where $U_{\text{T}} = \text{constant}$, which is independent of the parameter set as well.

Therefore, the relative entropy simplifies to

$$S_{\text{rel}} = \beta \langle U_{\text{CG}} \rangle_{\text{T}} - \beta F_{\text{CG}} + \text{constant} \tag{3.7}$$

where the constant will shift the relative entropy by some amount. The CG free energy, F_{CG}, is calculated during the multiparameter simulation, as described in Chapter 2. The calculation for the CG energy averaged over the target ensemble, $\langle U_{\text{CG}} \rangle_{\text{T}}$, is given below.

3.2.2 Efficient Calculation of $\langle U_{\mathrm{CG}} \rangle_{\mathrm{T}}$

The relative entropy depends on the average coarse-grained energy calculated over every configuration in the target ensemble, $\langle U_{\mathrm{CG}} \rangle_{\mathrm{T}}$. The calculation required information for every configuration of the target ensemble.

If the target is the experimentally determined native structure, then there is only one configuration, and the average energy in the target ensemble is just $E_{\mathrm{sc}}(r_{\mathrm{nat}})$ where r_{nat} is the native conformation.

However, if the target ensemble is the Molecular Dynamics simulation of the same protein, there are N_c configurations, and the CG energy must be calculated for each configuration. In order speed up the calculation of the ensemble average CG energy for either target ensemble, $\langle U_{\mathrm{CG}}(\overline{\lambda}) \rangle_{\mathrm{T}}$, we generated histograms of the separation distances, r, for each interaction pair, ij. The histograms were generated as a function of the two sequence indices, i and j, and was denoted $H_{ij}(r_k)$. Each histogram corresponds to the number of times the separation distance between the cα atoms at index i and j had a value in the range $[r, r+b]$, where b is the histogram bin size.

The bin size was chosen to be, $b = 0.001$, which was the degree of precision given in the raw data for the atom coordinates. Thus, the binned histograms did not significantly alter or compress the information.

The CG side chain interaction energy can be calculated from the histograms by summing over all interaction pairs, ij, and summing over each bin in the histograms, given as

$$\langle E_{\mathrm{sc}} \rangle_{\mathrm{T}} = \frac{1}{N_c} \sum_{i<j} b_i b_j \sum_{k}^{N_{\mathrm{bins}}} H_{ij}(r_k) \varepsilon_{\mathrm{sc}}(r_k) \tag{3.8}$$

where all of the terms are defined above. Using Eq. 3.8 avoids averaging over all conformations in the target ensemble, which speeds up the calculation of $\langle E_{\mathrm{CG}} \rangle_{\mathrm{T}}$.

3.2.3 Gradient of the Relative Entropy

The optimization schemes used here require the gradient of the relative entropy to be determined. The gradient of the relative entropy is written as,

$$\nabla_{\bar{b}} S_{\mathrm{rel}} = \frac{\partial}{\partial \bar{b}} S_{\mathrm{rel}} = \Big(\frac{\partial S_{\mathrm{rel}}}{\partial b_1}, \ldots, \frac{\partial S_{\mathrm{rel}}}{\partial b_{20}} \Big). \tag{3.9}$$

The gradient of the relative entropy with respect to the set of all CG parameters, λ_i, was found by Carmichael and Shell to be [26]

$$\frac{\partial S_{rel}}{\partial \lambda} = \beta \Big\langle \frac{\partial U_{CG}}{\partial \lambda} \Big\rangle_{AA} - \beta \Big\langle \frac{\partial U_{CG}}{\partial \lambda} \Big\rangle_{CG} \tag{3.10}$$

which can be simplified for the energy function given in Eq. 3.4 by noting that the gradient of the CG energy as a function of one of the amino-acid types, p, is

$$\frac{\partial U_{CG}}{\partial \lambda} = \frac{\partial E_{sc}(r_n)}{\partial b_p} = -\frac{\partial}{\partial b_p} \sum_{i<j} b_i b_j \varepsilon_{sc}(r_{ij}) \tag{3.11}$$

which can be solved to give

$$\frac{\partial E_{sc}(r_n)}{\partial b_p} = -\sum_{i<j} \delta_{ip} b_j \varepsilon_{sc}(r_{pj}) - \sum_{i<j} \delta_{pj} b_i \varepsilon_{sc}(r_{ip}) \tag{3.12}$$

where the delta function, δ_{ip}, means that only the side-chain interactions that involve the p-th amino-acid type are included in the energy calculation for the gradient with respect to the p-th amino-acid type.

It is important to note that for a function, f, that depends linearly on some system dependent property, $\overline{\lambda}$, the gradient is $\nabla f(\overline{\lambda}) =$ constant. This is the case here for the potential energy function, and thus, the relative entropy. The second derivative

Hessian matrix for a linear function is zero or positive, which means that the function, f, will have a basin shaped landscape with only one minimum. Therefore, determining the local minimum of the relative entropy via line minimization techniques should also determine the global minimum of the relative entropy.

3.2.4 Results for Target 1: Single Native Structure

As a first test, we applied our systematic CG method, as described in section 2.2, to determine the optimal model parameter set for the 13-parameter side-chain potential energy function. The relative entropy was minimized for Target 1, which was the experimentally determined native structure of the protein BBA. The line minimization procedure requires an initial parameter set and direction, which was chosen in the following way: the initial parameter set was a randomly generated guess, and the initial gradients were calculated from a fixed temperature simulation. Each multi-parameter simulation (corresponding to one iteration of the method) was run for 4 million Monte Carlo cycles, saving every 100^{th} configuration. The parameter set was discretized with 10 dynamic indices, and the steepest-descent method with a step size $\alpha = 0.01$ was used to generate the line in parameter space.

A simple way to visualize how the parameter set changed throughout the iterative process is to plot the value of each parameter for each iteration, as shown in Fig. 3.4. There are 13 model parameters that are changing, one for each of the 13 different amino acid types present in the protein BBA.

In principle, the theoretical converged parameter set would be the one for which the gradient of the relative entropy is zero for all parameters in the set. In practice, it is not expected that the gradient will ever be zero due to statistical fluctuations. Therefore, the converged parameter set is defined as the set for which the gradient of the relative entropy is sufficiently small, i.e $|\overline{g}| < \epsilon$.

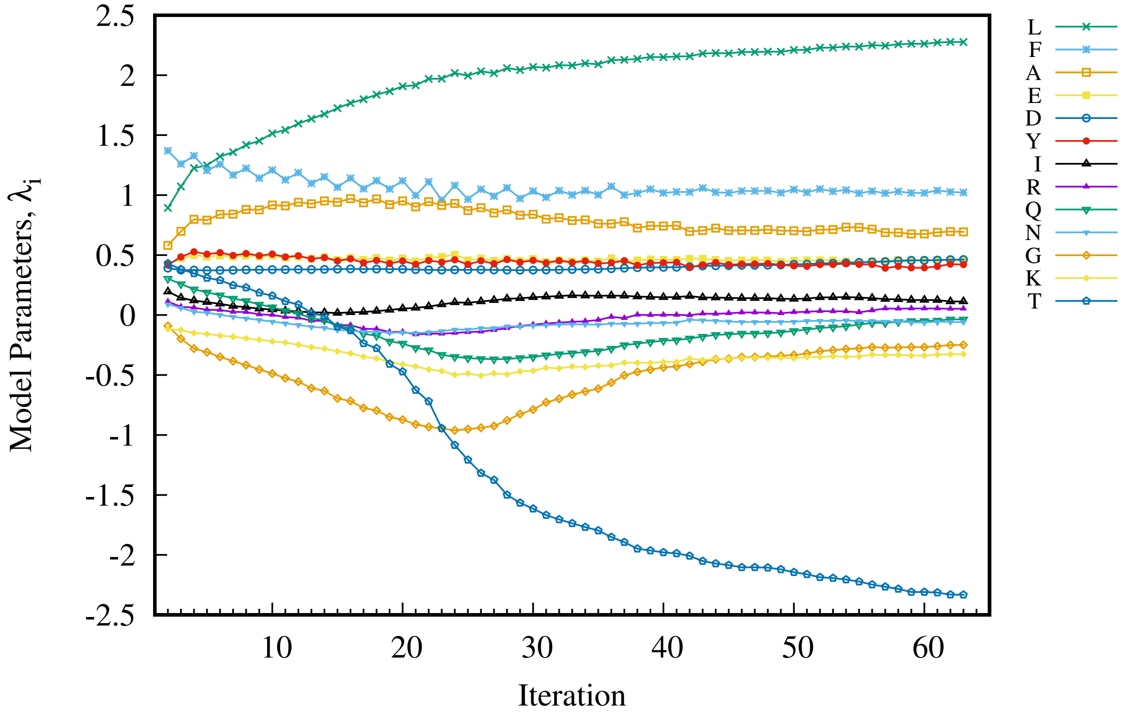

Figure 3.4: The value of each model parameter plotted versus the number of iterations. This graph shows how each of the model parameters changed throughout the simulations following the systematic coarse-grained method to determine the optimal parameter set.

The optimal parameter set, $\overline{\lambda}_{\text{opt}}$, is taken to be the parameter set at the final iteration in Fig. 3.4. We then determined the thermodynamic properties of the CG model with $\overline{\lambda} = \overline{\lambda}_{\text{opt}}$. This was done by carrying out additional conventional fixed temperature simulations and then calculating the RMSD between the generated CG ensemble and the experimentally determined native structure, 1FME [28].

To visualize how the RMSD differs between the initial and converged sets, a histogram is generated to show the probability that a CG ensemble has a RMSD in a small binned range. This gives a probability distribution, $P(\text{RMSD})$, of having a given RMSD value for each parameter set. This is shown in Figure 3.5.

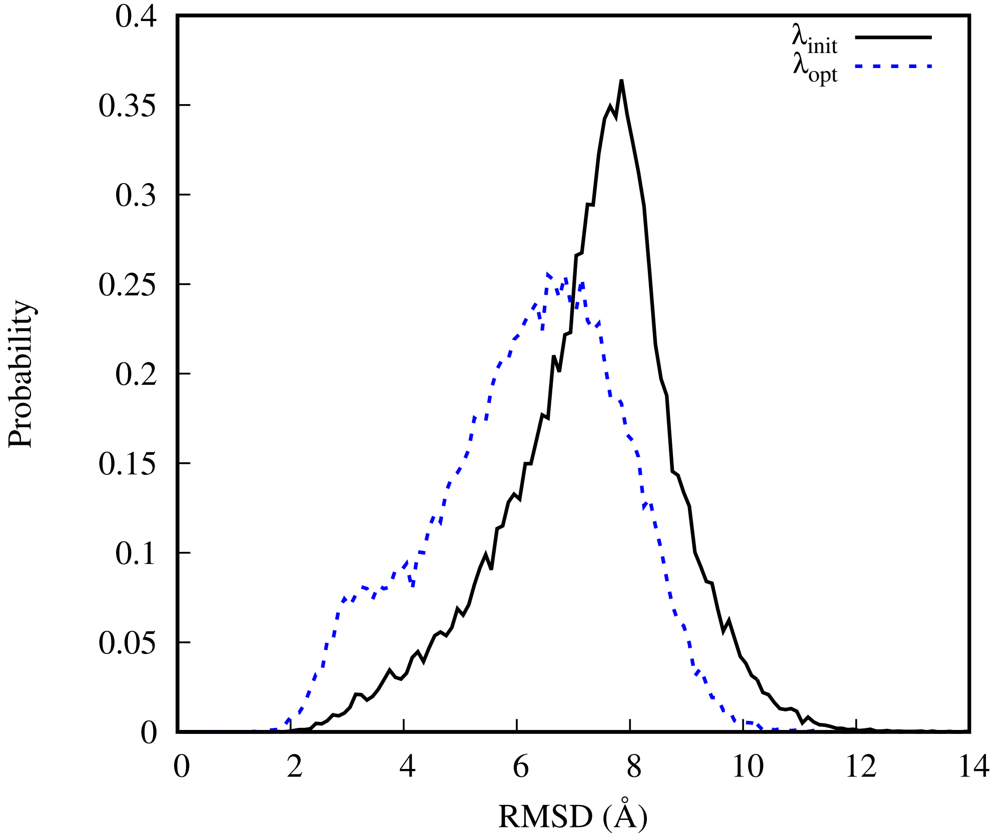

Figure 3.5: A histogram of the number of times a CG configuration has a RSMD value within a small binned range. The CG ensemble contained 40000 configurations, and the RMSD bin size was 0.05. The black curve is for the initial guess parameter set, λ_{init}, and the blue curve is for the final optimal parameter set, λ_{opt}.

Figure 3.5 shows a shift in the distribution towards lower values of RMSD. The probability distributions, $P(\text{RMSD})$, were obtained using the CG model with either the initial parameter set (black), or the optimized (blue) model parameter set. Due to the nature of coarse-graining, it was not expected that the RMSD would be zero, but it was expected that the converged parameter set should have a smaller RMSD if it properly captured the folded conformation. The figure above shows the converged set has two peaks, at 3.325Å and 6.875Å, while the initial parameter set has one peak at 7.865Å. The two peaks of the converged parameter set RMSD implies that there were a number of configurations that were folded into a configuration resembling the native state, but with a slightly larger RMSD around 3Å.

3.2.5 Results for Target 2: Molecular Dynamics Ensemble

In the second test, we applied our systematic method to the target ensemble from the Molecular Dynamics simulation described in section 3.1. The initial value for the parameter was chosen to be $b_0(a_i) = 0.3$ for all a_i, and the initial gradients were determined from a fixed temperature simulation.

For each iteration of the systematic CG method, we carried out four simulations with 8 million Monte Carlo cycles each. The gradients for the four simulations were averaged, and the average gradient was used as the direction for the steepest-descent method when setting up the next simulation.

The results for the optimal CG model parameters when using the MD simulation ensemble as the target are shown in Figure 3.6. The parameters converged to a parameter set different to that found when using the experimentally determined structure as the target, and the convergence was faster (fewer iterations).

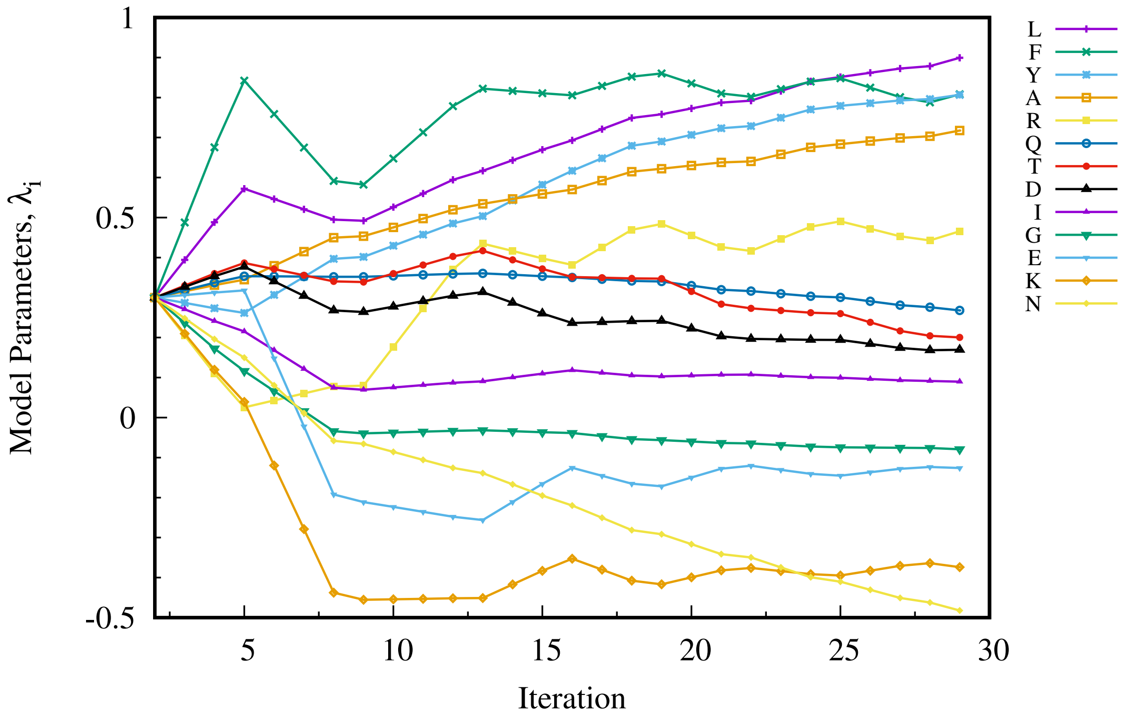

Figure 3.6: The value of each model parameter, λ_i, plotted versus the number of iterations required for the systematic CG method to find an optimal parameter set. These results are for the set of simulations that used the ensemble generated from the long MD simulation as the target for the relative entropy minimization.

As shown in Figure 3.7, the $P(\text{RMSD})$ obtained using the converged CG parameter set, $\overline{\lambda}_{\text{opt}}$, does not match the results for $P(\text{RMSD})$ for the MD simulation target. In fact, the converged parameter set does not show a second peak in the RMSD, which implies the CG ensemble did not sample folded and unfolded states like the MD simulation did. It does, however, match the target ensemble at higher values of RMSD ≥ 9Å.

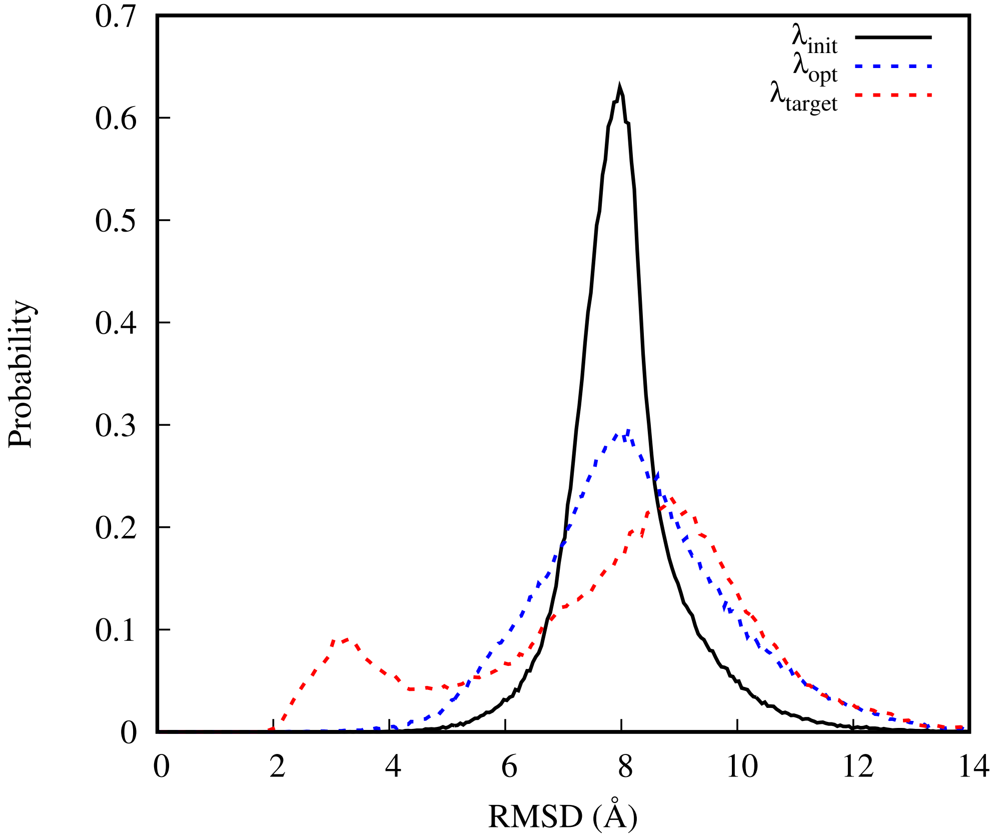

Figure 3.7: A histogram of the number of times a CG configuration has a RSMD value within a small binned range. The CG ensemble contained 320000 configurations, and the RMSD bin size was 0.05. The black curve is for the initial guess parameter set and the blue curve is for the final converged data set. The red curve is for the RMSD probability histogram for the MD simulation ensemble.

3.2.6 Optimal Choice of Step-Size α

Figure 3.6 showed that there were many times when multiple iterations were required before a minimum was found along a given direction. This occurs when the discretized set of parameters does not extend far enough into parameter space to include the minimum. The systematic CG method used here would be more efficient if a minimum was found after each iteration, which would occur if the length of the line in parameter space was sufficiently long.

The systematic CG method allows the parameter set to be discretized according to

$$\overline{\lambda}_i = \overline{\lambda}_0 + i\alpha\overline{g} \quad \text{for} \quad i = 0, 1, \ldots, K-1 \tag{3.13}$$

where $\overline{\lambda}_0$ is the initial parameter, and $\overline{g}$ is the direction, and α is the step-size. It is clear from this equation that changing α or K will change the range of the line in parameter space.

Increasing the number of discrete values would require longer simulations in order to sample each discrete state the same number of times (obtain the same statistics). However, increasing the step size when generating the discrete parameter sets does not change the number of times each state is sampled, and thus does not affect the number of MC cycles required. Increasing the step size could affect the ability of the multiparameter simulation to visit each discrete state equally. This issue is overcome by tuning the initial control parameter dependent function, $h(\overline{\lambda})$. Since this function is used in the acceptance criteria for the multiparameter simulation, it directly affects how probable it is to sample each state. Therefore, by making a better guess for that function, the simulation can be set up to evenly sample every state.

In Figure 3.6, the step size was too small for the discretized parameter set to sample the state corresponding to the relative entropy minimum. In other words, the parameter set $\overline{\lambda}_{K-1}$ comes before the optimal parameter set, $\overline{\lambda}_{\text{opt}}$, in the direction $\overline{g}$. In order to determine what the optimal step size was, two separate tests were run for different step sizes.

The first test was for $\alpha = 0.05$, which is a factor of 5 larger than the previous step size. This was done based on the fact that for the previous results, some iterations required up to 4 simulations before finding a minimum. The results for how the values of the parameter set changed for each iteration are shown in Figure 3.8, where the iteration number corresponds to a single simulation. This iterative approach using a step size of $\alpha = 0.05$ was labelled as "efficient" since each iteration required only one simulation, and the converged parameter set was obtained in only 15 iterations.

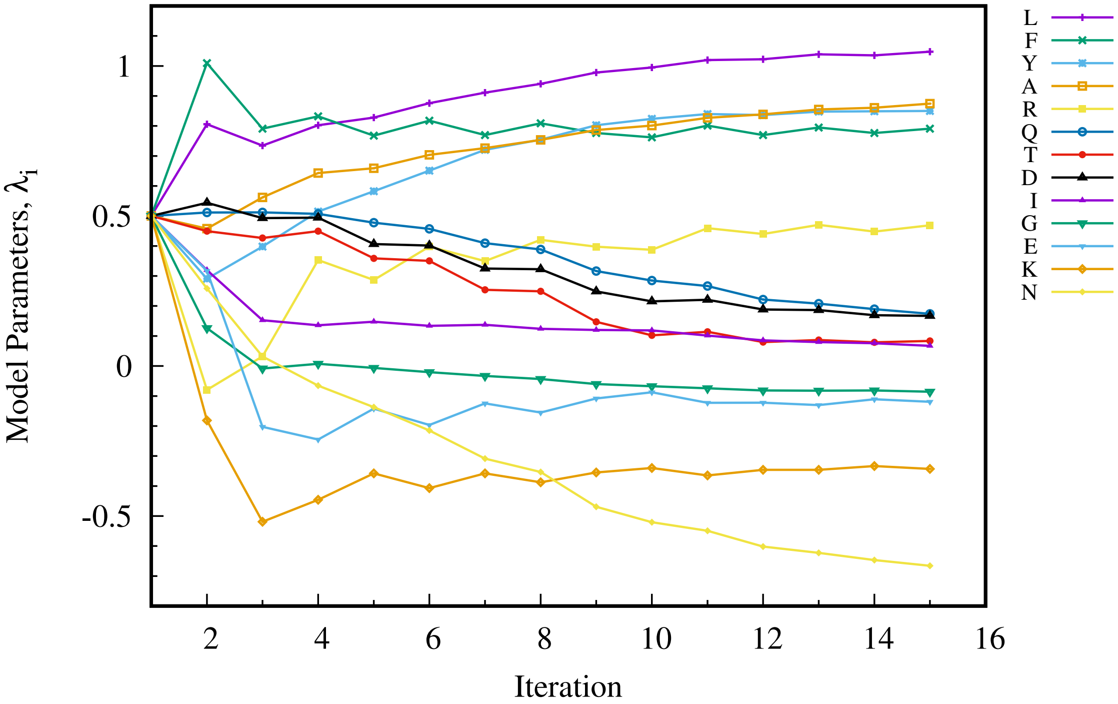

Figure 3.8: The value of each model parameter, λ_i, plotted versus the number of iterations required for the systematic CG method to find an optimal parameter set. These results are for the set of simulations that used the "efficient" step size choice of $\alpha = 0.05$. The target for the relative entropy minimization was the ensemble generated from the long MD simulation.

It is also important to note that the initial parameter set, $\overline{\lambda}_0 = 0.5$ for all parameters. Again, the initial gradients were determined from a separate fixed temperature simulation. All other simulation parameters were kept the same as those used to obtain Fig. 3.6. Thus, the converged parameter set for the two runs could be directly compared. The two graphs, Fig. 3.6 and Fig. 3.8, show that the step size and the initial guess do not affect the converged parameter set, as the two graphs both converged to the same values. The choice of step size does obviously affect the efficiency of the iterative approach, as demonstrated. However, it is not expected that the initial guess for the parameter set will have a significant affect on the efficiency of the process, as the method of finding the minimum relative entropy is largely controlled by the magnitude of the gradient.

The Cα-RMSD for the "efficient" step size choice was generated, but gave results almost identical to those in Fig. 3.7, since the converged parameter set was almost identical.

A final set of simulations were run with the goal of giving an indication that the above results did not converge to a local minimum instead of the global minimum. Since the optimization was done using steepest-descent method, which is a local line minimization technique, the converged set could be a local minimum. To demonstrate that the converged set was not stuck in a local minimum, the step size was increased again to $\alpha = 0.10$, which was 10 times larger than the initial step size, and twice as large as the "efficient" step size. This was done such that the process of discretizing the parameters covered a larger range of parameter space for each simulation.

The results for the iterative approach with $\alpha = 0.1$ for the parameter values as a function of number of iterations are shown in Figure 3.9. The converged parameter set was found after 17 iterations, and all parameters converged to the same results as above. Therefore, this shows that the converged parameter set is most likely in a global relative entropy minimum, and the choice of step size does affect the efficiency of the method, but a step size of $\alpha = 0.05$ is a good choice.

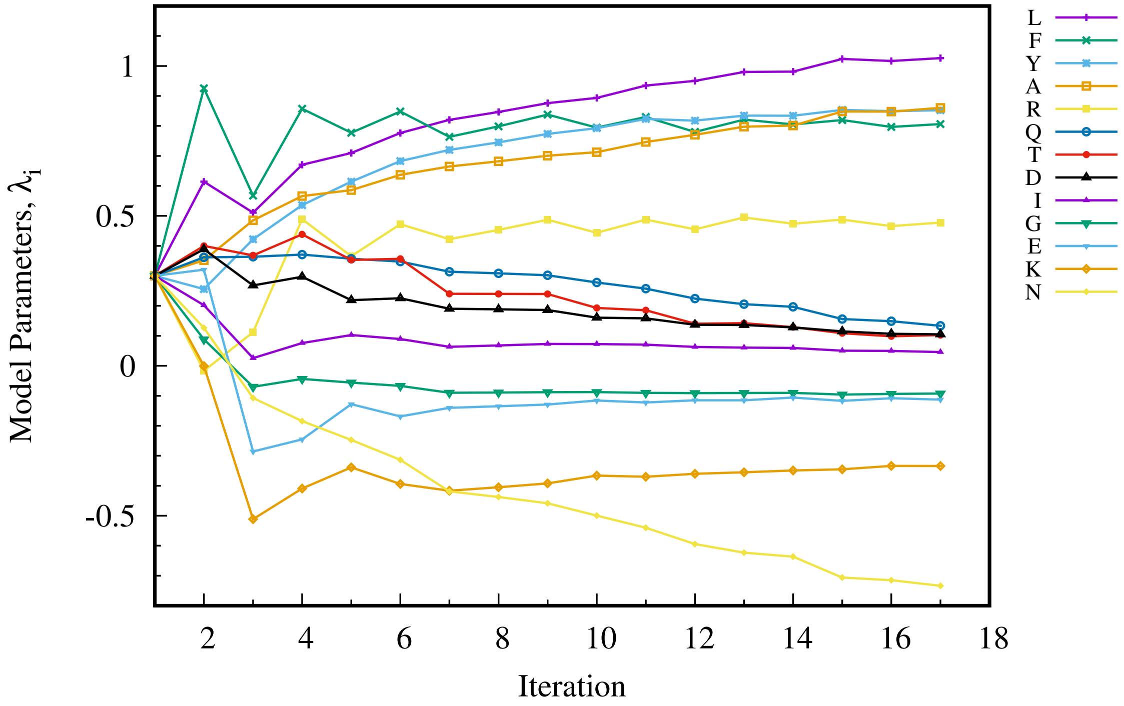

Figure 3.9: The value of each model parameter, λ_i, plotted versus the number of iterations required for the systematic CG method to find an optimal parameter set. These results are for the set of simulations that used the larger step size choice of $\alpha = 0.1$. The target for the relative entropy minimization was again the ensemble generated from the long MD simulation.

3.2.7 Comparing Optimal Model Parameters

Next we investigate how the $\overline{\lambda}_{\text{opt}}$ parameter sets for target 1 and target 2 compare. To do this, the final values for all of the 13 parameters are plotted on the same graph, in Figure 3.10. If the 13 parameters converged to the same results for either target ensemble, they would fall on the line $y = x$, which is plotted as reference.

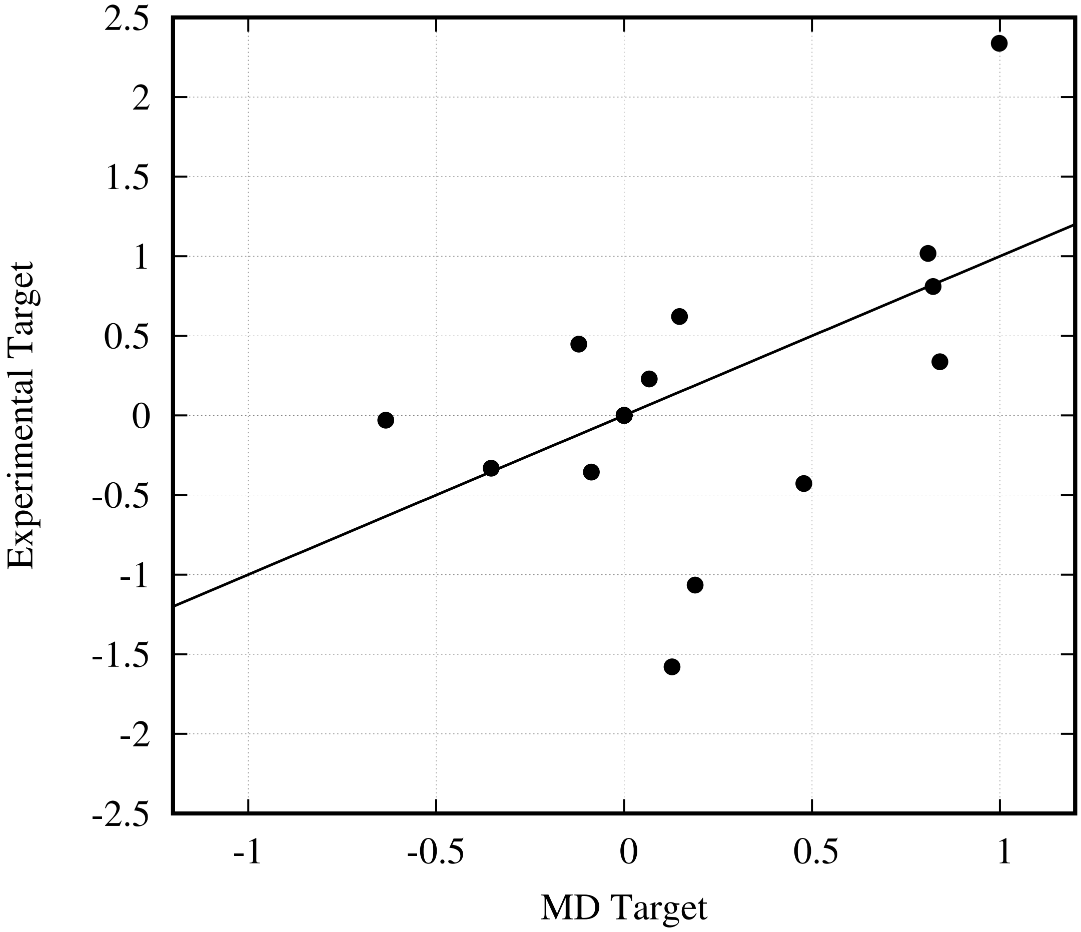

Figure 3.10: The optimal CG model parameters found using the systematic method for each of the two target ensembles are plotted here. The line $y = x$ shows where the values would lie if the optimal parameters were the same for the two targets. The single experimentally determined native structure target is on the y-axis, and the target from the ensemble from the MD simulation is on the x-axis.

Clearly, the converged set with the target being the single experimentally determined structure is not the same as when the target is a MD simulation ensemble. However, assuming there is one correct optimal CG parameter set, the systematic method should find the optimal parameter set regardless of the choice of the target ensemble.

3.2.8 Conclusion

The results presented above show that the systematic coarse-grained method for determining the optimal model parameter set, $\overline{\lambda}_{\text{opt}}$ works to some extent. The method was able to systematically converge to a parameter set from some initial guess parameter set. In the case with the experimentally determined single native structure as the target, the convergence was very slow, taking over 60 iterations, but the converged set showed better RMSD results. Figure 3.5 showed a shift in the peak to a lower RMSD value, and a second weaker peak in the RMSD histogram at 3.32Å for the converged parameter set. This means that the converged parameter set was a better match to the target ensemble, and although the ensemble sampled a majority of states with an RMSD between 6 and 7 Å's, there were a number of configurations with structure similar to the target native state.

However, the results for the simulations using the ensemble generated from the D. E. Shaw Molecular Dynamics simulation as the target were not able to find a good optimal parameter set. The optimal parameter set was found multiple different ways, using different step sizes or initial guesses, but the converged parameter set was not able to match the results, such as $P(\text{RMSD})$, for the experimental or MD target. The method did allow for a systematic approach to finding some converged parameter set, and it was found that the size of step used in discretizing the parameter set allowed for faster, more efficient convergence. Further, it was determined that the initial guess parameter appears to have little affect on the rate of convergence, and the method is able to find the optimal parameter set regardless of handful of initial condition used here.

Therefore, the systematic CG method for determining the optimal model parameter set was shown to be an efficient and systematic way to determine the optimal parameters through minimizing the relative entropy between a CG ensemble and a target ensemble. However, the issue here was that the converged parameter set was not able to exactly match the experimental or all-atom simulation results in terms of folding properly, as shown in the RMSD histograms. One explanation for this issue is that the energy function used for the side-chain interaction was too simple. The systematic method was able to obtain the best possible converged parameter set, but since the functional form of the interaction energy had a very simple form, the optimal parameter set was still not enough to capture all the details of the side-chain interaction.

3.3 91-parameter Side-Chain Potential Energy Function

To test the prediction that the side-chain interaction energy above was not sufficient to capture the details of the side-chain interaction, we propose a better interaction energy term with more parameters. Here, we propose a 20 by 20 matrix $M(a_i, a_j)$, where the indices represent amino acid types, a_i and a_j. The model parameter set will be the elements of this matrix, where the elements represent an interaction strength between amino acid type a_i and a_j. Since M is symmetric (i.e. $M(a_i, a_j) = M(a_j, a_i)$) the number of unique matrix elements is $N(N-1)/2 + N$, so there are 210 unique parameters for $N = 20$ amino acid types.

However, since there are only 13 amino acid types present in the BBA protein being studies, the number of model parameters is less. The parameter set can be represented by a 13 by 13 matrix, which will have 91 unique elements. Therefore, the model parameter set that is being optimized is a vector with 91 components given as $\overline{\lambda} = M(a_i, a_j)$ for $i, j = 1, \ldots, 13$ and $i \geq j$. The side-chain interaction energy for configuration, $\overline{r}$ and parameter set, $\overline{\lambda}$, was given in Eq. 3.2.

Targets 1 and 2 defined above are used here as well, and the systematic CG method as described in Chapter 2 is followed here.

3.3.1 Results for Target 1: Single Native Structure

The initial parameter set was chosen to be $\overline{\lambda}_0 = 0.3$ for all 91 elements of the matrix, $M(a_i, a_j)$, and the initial gradients were determined from a fixed temperature simulation. The multiparameter simulation was then run 4 times with 8 million MC cycles each, and every 100th configuration was saved. The efficient step size $\alpha = 0.05$ that was determined earlier was used here as well.

Figure 3.11 shows that the majority of the parameters converge within the first 15 iterations, while some parameters take longer to converge. The interaction pairs with a value $\overline{\lambda} \geq 2$ are T-T, F-G, F-L, and L-G (strongly attractive), the interaction pairs with a value $\overline{\lambda} \leq -2$ are R-G, R-I, and Y-T (strongly repulsive).

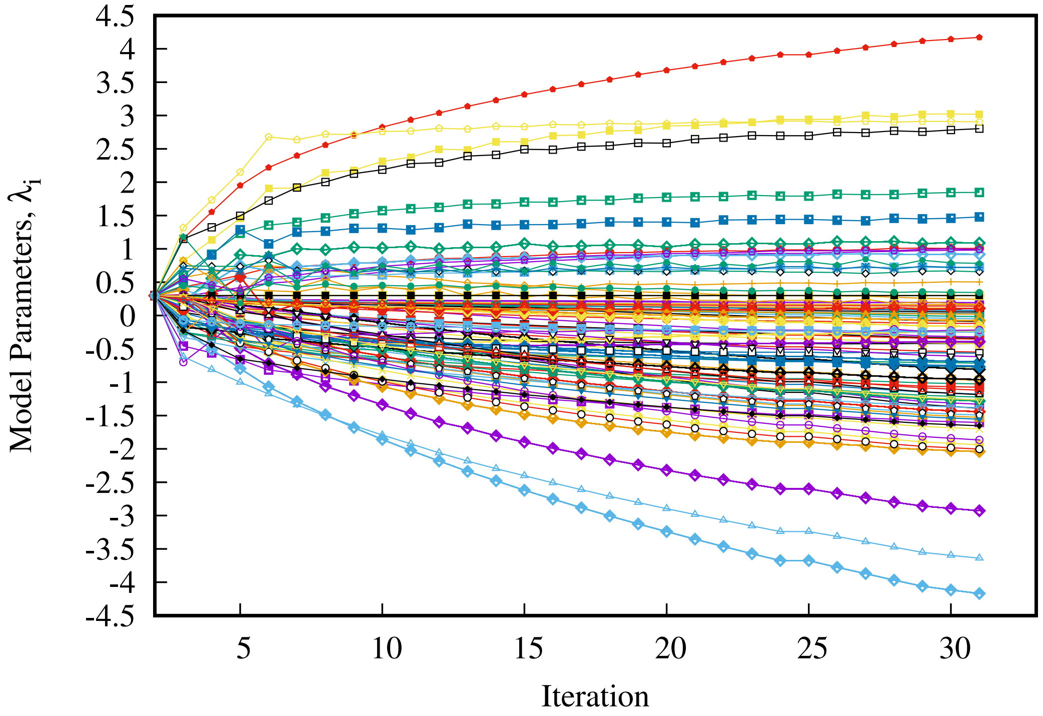

Figure 3.11: The value of each model parameter, λ_i, plotted versus the number of iterations required for the systematic CG method applied to the 91-parameter side-chain potential energy and with target 1, the experimentally determined native structure.

As before, the optimal parameter set was used to generate an ensemble of configurations from a fixed temperature simulation in order to obtain the RMSD distribution, $P(\text{RMSD})$. Unlike the simulations shown above with the 13-parameter side-chain interaction energy, the optimal set of 91 model parameters gives RMSD results that are much closer to the expected results. Shown in Figure 3.12, the converged parameter set has a larger probability of being in a configuration with RMSD between $2 - 4$Å, and the peak in the RMSD histogram occurs at 2.575Å. This result means that the optimal parameter set, $\overline{\lambda}_{\text{opt}}$, found by minimizing the relative entropy between the CG ensemble and the experimentally determined structure, prefers to be in a folded state very close to the true native state. Therefore, from a random initial guess for the 91

CG model parameters, the optimal parameter set obtained is capable of recapturing properties of the physical system. This was the goal of the systematic coarse-grained method, and it is shown here that the method works as expected for this choice of potential energy function and target ensemble.

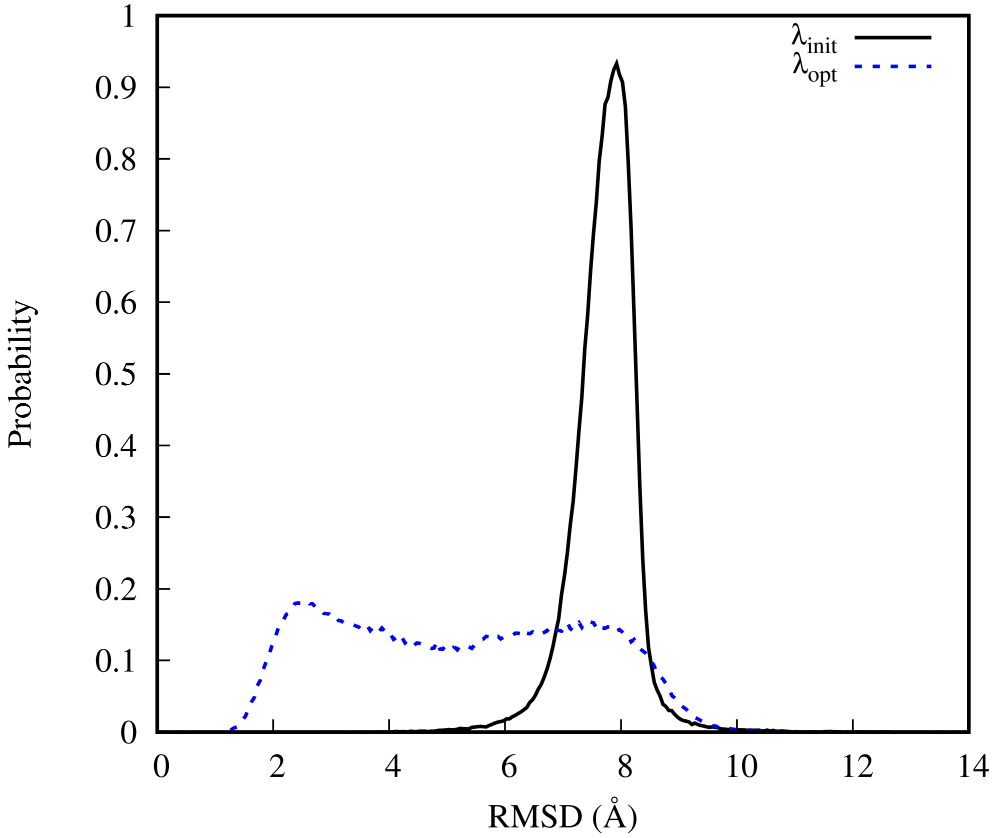

Figure 3.12: A histogram of the number of times a CG configuration with the matrix parameter set has a RSMD value within a small binned range. The CG ensemble contained 320000 configurations, and the RMSD bin size was 0.05. The black curve is for the initial guess parameter set and the blue curve is for the final converged data set.

3.3.2 Results for Target 2: Molecular Dynamics ensemble

The next set of simulations were done to test how well the optimal parameter set could be found when using the ensemble generated from the Molecular Dynamics simulation as the target for the relative entropy minimization. Again, 4 multiparameter simulations with 8 million MC cycles were run and the iterative approach was followed. The step size was decreased to $\alpha = 0.03$ instead of $\alpha = 0.05$. The initial parameter set was chosen to be $\overline{\lambda}_0 = 0.0$ and the initial gradients, $\overline{g}_0$, were found from a fixed temperature simulation.

Figure 3.13 shows that the majority of the model parameters converged in fewer than 10 iterations, and the optimal model parameter set, $\overline{\lambda}_{\text{opt}}$ was taken from the 20^{th} iteration.

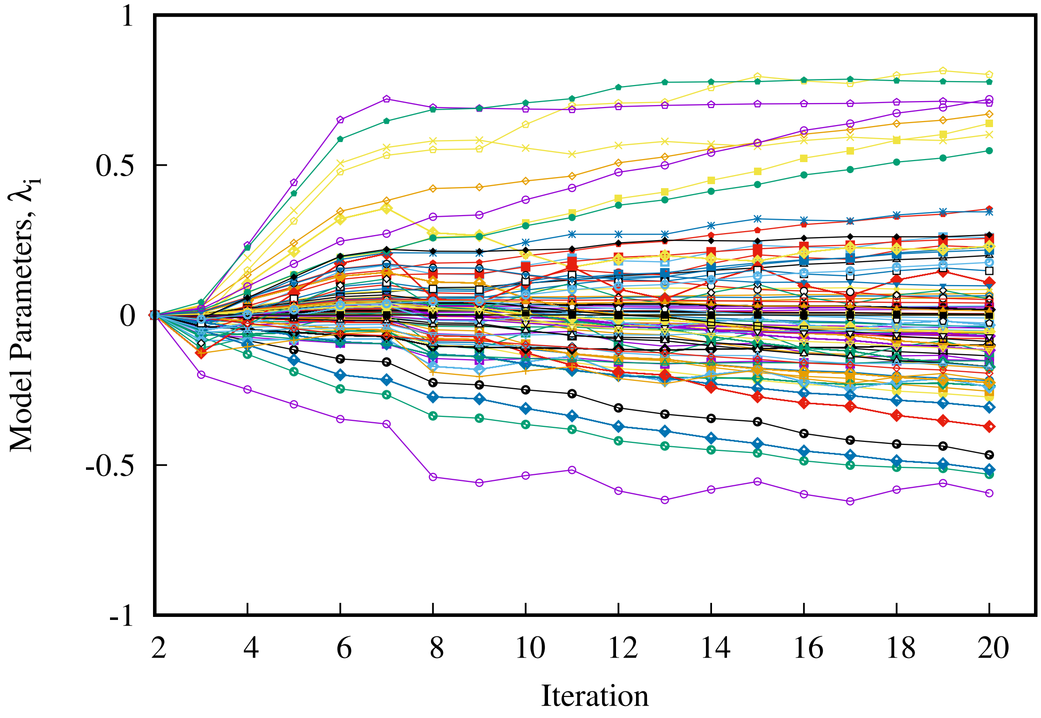

Figure 3.13: The value of λ_i versus the number of iterations for the set of simulations that used the new side-chain interaction energy with a matrix for the parameter set. The MD ensemble was the target for the relative entropy minimization, and steepest-descent minimization method was used to determine successive parameter set.

Again, the optimal parameter set was used to generate an ensemble of configurations from a set of 4 fixed temperature simulation, and the probability distribution,

P(RMSD), was calculated for the optimal set, $\overline{\lambda}_{\mathrm{opt}}$. The RMSD distribution for the optimal set of 91 parameters is compared to the target ensemble distribution obtained from the all-atom MD simulation, which is given in Fig. 3.3. The two distributions were also compared to the distribution for the initial guess parameter set. The results for the RMSD distributions are shown in Fig. 3.14.

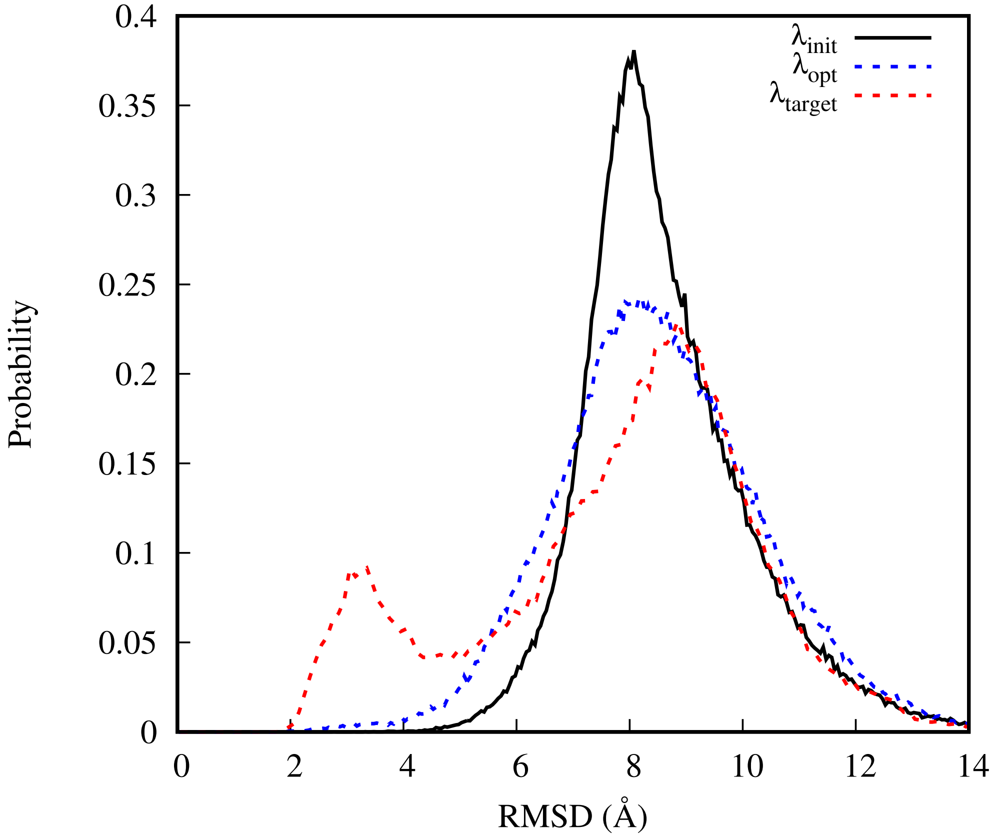

Figure 3.14: The RMSD histogram for the CG configuration with the matrix parameter set. Target ensemble was the MD simulation ensemble, and steepest-descent line minimization was used to find successive parameter sets. The black curve is for the initial guess parameter set and the blue curve is for the final converged data set.

Figure 3.14 shows that the optimal parameter set (blue line) does not capture the bimodal behaviour of the target distribution (red line), corresponding to the folded and unfolded regimes. The optimal parameter set is able to capture RMSD distribution for the unfolded states between $6 - 12$Å, but there are very few configurations that are in a folded state close to the native structure. The inability to recreate the peak corresponding to the folded state, ~ 3Å, could be due to the fact that the all-atom simulation is dominated by configurations in the unfolded state, or the optimal parameter set is not in a global minimum of the relative entropy, S_{rel}. Here, the systematic CG method found a parameter set that was optimized to the unfolded region of the all-atom target. This result leads to the important conclusion that the ability of the systematic CG method to determine the correct optimal parameter set depends on the choice of the target ensemble.

3.3.2.1 Testing Line Minimization Schemes

The systematic CG method relies on a line minimization technique to determine the directions to travel in parameter space. The two different minimization techniques looked at here are steepest-descent and conjugate-gradient, and results for both were shown in the validation of the method in Chapter 2. Conjugate gradient method is expected to be more efficient at searching through parameter space when the landscape is smooth and parabolic in shape. For the case of the protein folding problem explored here, the relative entropy landscape appears smooth in the sense that it does not contain multiple minima. Therefore, the conjugate gradient minimization technique should be a more efficient way to search through parameter space and find the optimal parameter set. This was tested by applying the systematic CG method to same initial conditions that were used to create Fig. 3.13, but using the conjugate gradient method to determined the direction for each new line.

Figure 3.15 shows the model parameter convergence when the conjugate-gradient method was used to determine the direction to travel for successive parameter sets. In contrast to Fig. 3.13, the convergence appears much smoother, which represents the fact that each new direction calculated using the conjugate gradient method is moving the model parameters directly in the direction of the optimal set. Furthermore, the convergence occurs in just 13 iterations, as opposed to the 20 iterations that were required for the case of the steepest descent minimization.

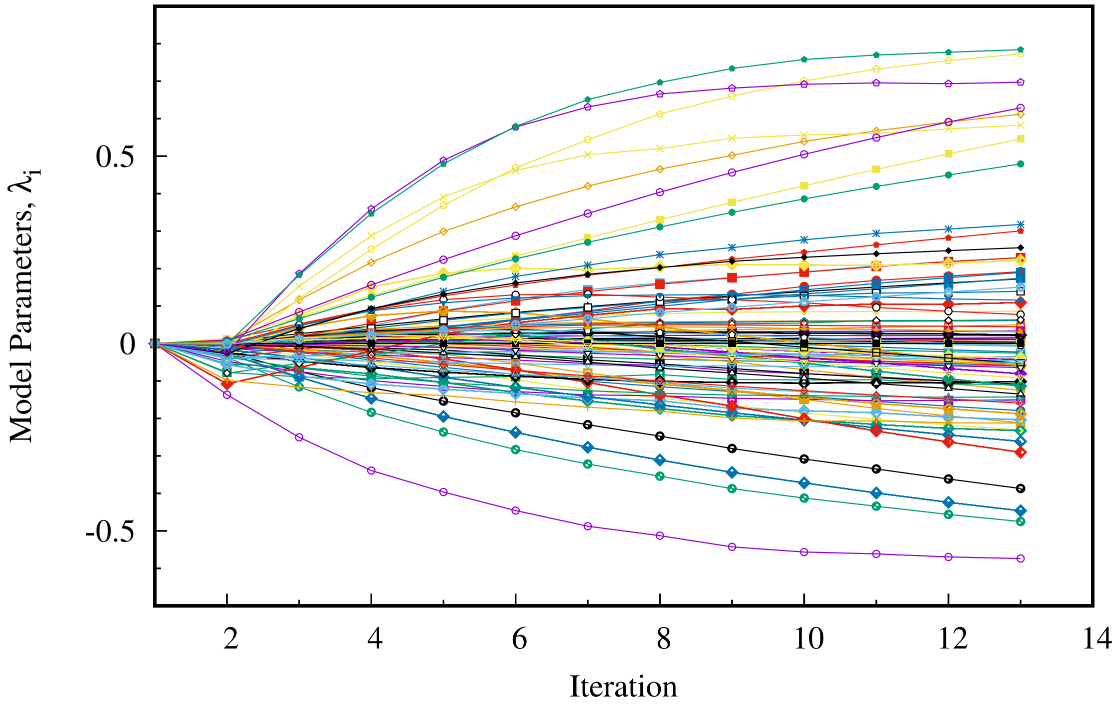

Figure 3.15: The value of λ_i versus the number of iterations for the set of simulations that used the new side-chain interaction energy with a matrix for the parameter set. The MD ensemble was the target for the relative entropy minimization, and conjugate-gradient minimization method was used to determine successive parameter set.

It is clear by comparing Figures 3.13 and 3.15 that the conjugate gradient method is more efficient in determining the optimal parameter set. However, both minimization schemes converge to the same parameter set within a small uncertainty, which means that the issue of the RMSD distribution not matching the target distribution is still relevant here. The RMSD distribution graph for the conjugate gradient case is not shown here because it is the same as Fig. 3.14 above.

3.3.3 Comparing Optimal Model Parameter

The scatter plot showing how the optimal parameter sets for the two targets compare to each other is shown in Fig. 3.16. The range of the interaction strengths for the experimental target (y-axis) is $b(a_i) \in [-5, 5]$, which is a larger spread (more attractive or repulsive) than the 13-parameter potential energy function. While some of the parameter values fall close to the line $y = x$, a significant number of the parameters are far from that line. The disagreement in the optimal parameter sets for the two target ensembles can be understood by considering the unfolded state in the MD target ensemble since the behavior of the unfolded state can be recreated without many strong interactions. It is this disagreement that gives rise to the difference in the RMSD distribution graphs (Fig. 3.12 and 3.14).

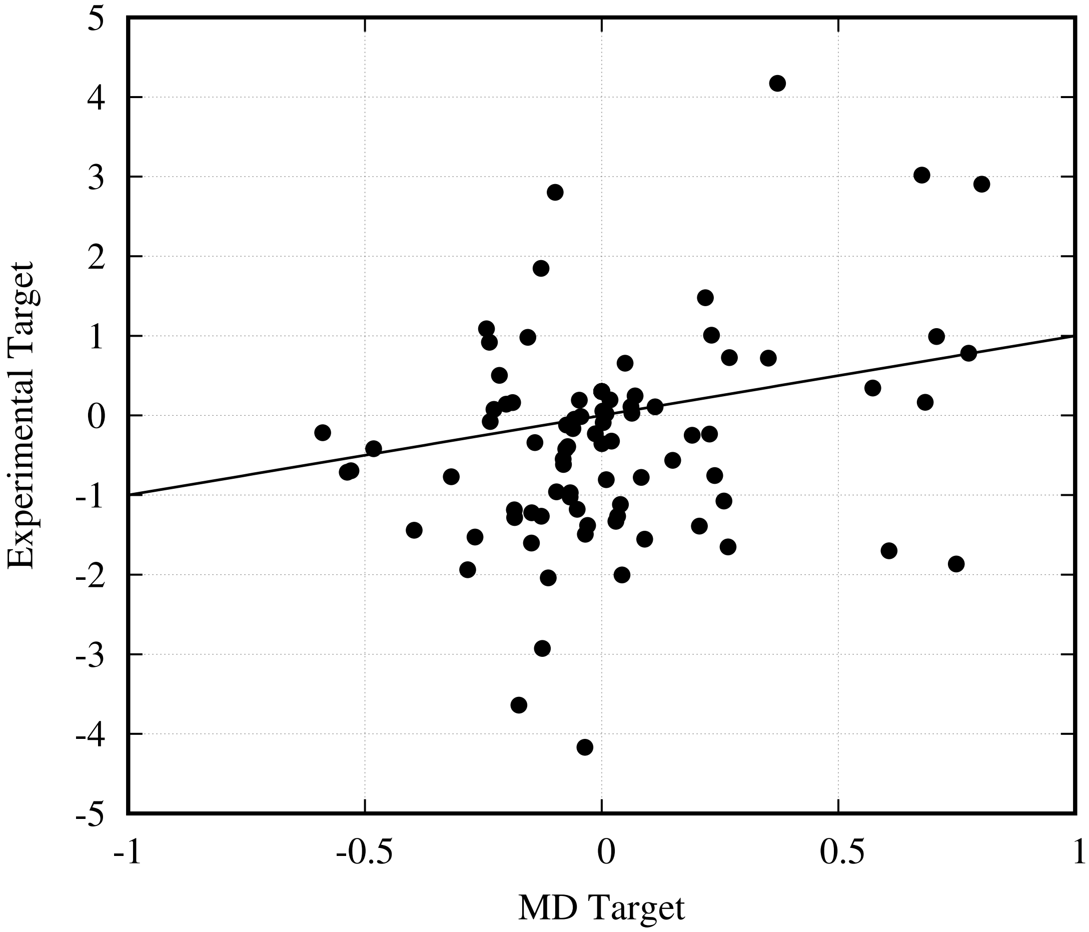

Figure 3.16: The optimal CG model parameters found using the systematic method for each of the two target ensembles are plotted here for the case of the 91-parameter potential energy function. The line $y = x$ shows where the values would lie if the optimal parameters were the same for the two targets.

3.3.4 Conclusion

The 91-parameter potential energy function studied here was better than the 13-parameter function in two ways: The optimal parameter set was found in fewer iterations for both targets, and the optimal parameter set for the experimentally determined single structure target allowed the protein to sample the folded state the majority of the time.

All of the results shown so far, for both potential energy functions and both target ensembles, show that the systematic CG method described in Chapter 2 is capable of determining the optimal parameter set. However, the degree to which the optimal parameter set allows the CG ensemble to match the physical system directly depends on the choice of the parameter set and target function. A parameter set with more free parameters, and a very precise target ensemble give the best results for the systematic CG method.

3.4 Global Optimization Approach

The iterative approach used for the systematic CG method for determining the optimal CG model parameter set described in Chapter 2 has one main drawback, which is that the method is based on consecutive line minimizations. This means that the entire iterative approach cannot be described as a global optimization in parameter space, and thus the optimal parameter set could potentially be stuck in a local minimum. In the above iterative approach, if the relative entropy landscape is rough over the parameter space, there could be many local minima and one global minimum. This could cause the iterative line minimizations to find a minimum that is not the global minimum.

Many global optimization methods have already been developed for common potential energy functions in molecular simulations [30]. Here, we propose a global approach that relies on two observations from the above results. First, while the systematic CG method has been applied to a discrete set of points, λ_i, along a line in parameter space, this is not required. In fact, simulations of the probability distribution in Eq. 2.2 can, in principle, be done for any set of points $\overline{\lambda}$. Second, we note that by making the choice $h(\overline{\lambda}) = \beta\langle U_{\mathrm{CG}}\rangle_{\mathrm{T}}$, the marginal distribution becomes

$$p(\overline{\lambda}) \propto e^{-\beta F(\overline{\lambda})+\beta\langle U(r,\overline{\lambda})\rangle_{\mathrm{T}}} \propto e^{S_{\mathrm{rel}}(\overline{\lambda})}. \tag{3.14}$$

The marginal distribution is proportional to the exponential of the relative entropy, which means the most probable states will be those with the highest relative entropy. However, the relative entropy was a measure of the difference between two probability distributions, and a large relative entropy means the two distributions do not match. Practically, that means that the parameter set that corresponds to a higher $S_{\mathrm{rel}}(\overline{\lambda})$ is further from the optimal, $\overline{\lambda}_{\mathrm{opt}}$. Therefore, simulating the marginal distribution given in Eq. 3.14 will tend to sample the worst parameter sets.

The method here is to run a multiparameter simulation with the marginal distribution described here and systematically remove parameter sets that are sampled frequently during the simulation (i.e. parameter sets with large $S_{\mathrm{rel}}(\overline{\lambda})$). This process will lead to one of two outcomes: only one parameter set, $\overline{\lambda}_{\mathrm{opt}}$, remains after all of the highest probable states are removed, or a few parameter sets remain with a roughly uniform probability distribution. If it is the second of the two outcomes, a single multiparameter simulation can be done like before for the remaining parameters, and the relative entropy can be calculated and the optimal parameter set will be obtained.

This process eliminates the need to have an initial guess for the control function, $h(\overline{\lambda})$, as it will be initially equal to the ensemble average of the CG energy function calculated in the target ensemble, $h(\overline{\lambda}) = \beta\langle U(r,\overline{\lambda})\rangle_{\mathrm{T}}$. Furthermore, this method can be expanded to be an iterative process while still remaining a global optimization approach.

3.4.1 Iterative Approach using Global Optimization

One possible iterative approach to determine the optimal parameter set, $\overline{\lambda}_{\text{opt}}$, using the global optimization scheme described above is as follows:

1. Choose initial parameter set, $\overline{\lambda}_0$
2. Generate N different parameter points, $\overline{\lambda}_i$, in parameter space
3. Run the multiparameter simulation with $h(\overline{\lambda}) = \beta\langle U_{\text{CG}}\rangle_{\text{T}}$. Record $p(\overline{\lambda}) \propto e^{S_{\text{rel}}(\overline{\lambda})}$ and successively remove $\overline{\lambda}_i$'s with highest $p(\overline{\lambda})$ to find the best parameter set, $\overline{\lambda}_{\text{best}}$ (last remaining set)
4. Repeat steps 2 and 3, where each new set of parameters relies on $\overline{\lambda}_{\text{best}}$, until $\overline{\lambda}_{\text{opt}}$ is found

3.4.2 Testing the Iterative Global Optimization Approach

The iterative approach for the global optimization scheme was tested with the 91-parameter potential energy function, using the single experimentally determined native structure as a target for the calculation of $h(\overline{\lambda}) = \beta\langle U(r, \overline{\lambda})\rangle_{\text{T}}$.

The initial N parameter points are generated using the Box-Muller transformation method, which generates random numbers that satisfy a Gaussian distribution from another random number generator that samples a uniform distribution. The Box-Muller method takes two uniformly distributed random numbers, R_1 and R_2, and generates two Gaussian distributed random numbers centred at zero with a variance of 1 [31]. The Box-Muller transformation equations are,

$$Z_1 = \sqrt{-2\ln R_1}\cos(2\pi R_2)$$

$$Z_2 = \sqrt{-2\ln R_1}\sin(2\pi R_2)$$

Now, a random number with a Gaussian distribution centred at μ with standard deviation, σ, can be generated using

$$X = Z_1\sigma + \mu. \tag{3.15}$$

Figure 3.17 shows the distribution of two parameters using the Box-Muller transformation to generate a Gaussian distribution with a given center point and standard deviation, σ.

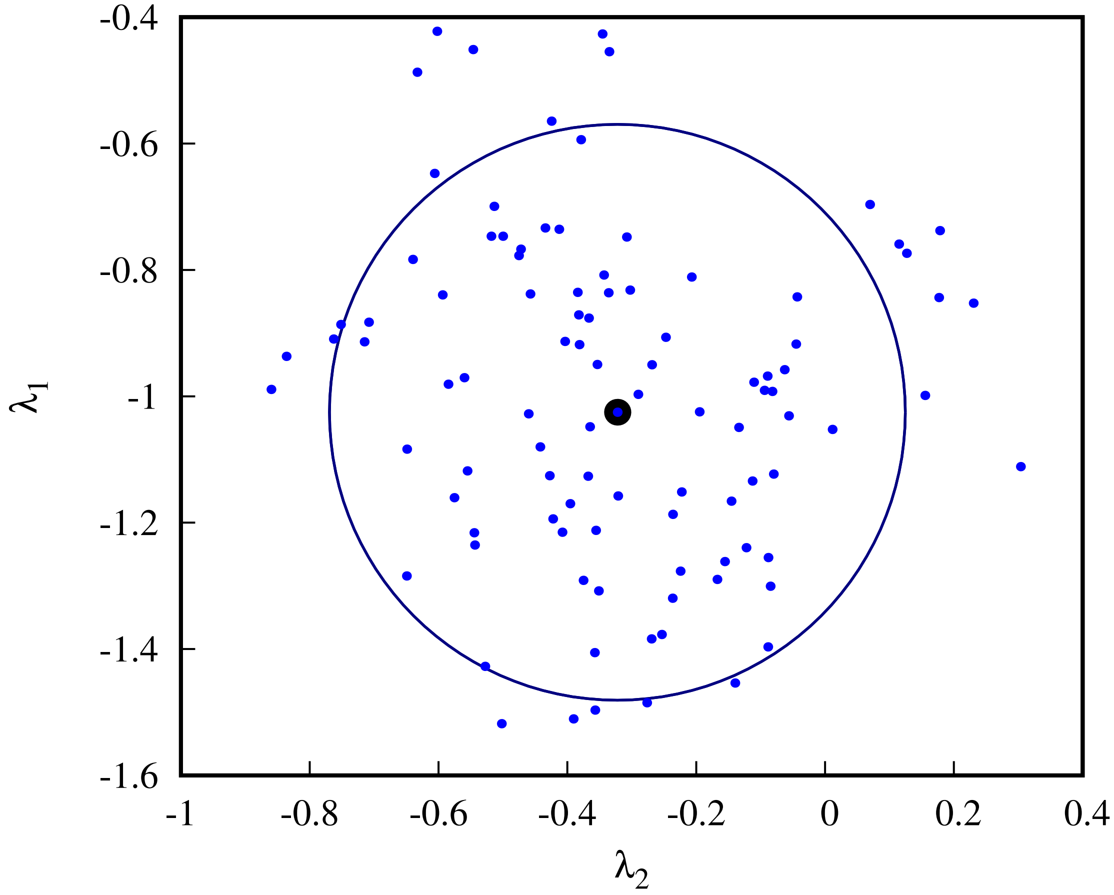

Figure 3.17: The parameter sets, λ_1 and λ_2, have a Gaussian distribution centred at the solid black circle, and spread in parameter space with a standard deviation, $\sigma = 0.25$, which is represented by a circle of radius $r = \sigma$.

The $N = 100$ parameter points were centred around the optimal parameter set, $\overline{\lambda}_{\text{opt}}$, that was found from the systematic CG method for the same potential energy and target (Fig. 3.11). The standard deviation was chosen to be $\sigma = 0.25$ such that the subset of parameter space was large, but the difference in energy between the center and the outermost parameter sets was not enormous. The multiparameter simulation was run for 10 million MC cycles, and the most frequently sampled state was removed every 100,000 MC cycles, therefore, one parameter set would remain after the simulation.

Figure 3.18 shows how the index of the parameter changes throughout the multiparameter simulation. For roughly the first 1 million MC cycles, the system gets stuck in particular states, and does not fluctuate (represented by a small horizontal line). The sharp vertical lines represent the system being kicked out of a given state due to that parameter set being removed. When that happens, the system randomly jumps to another state that has not been removed, and samples the remaining states.

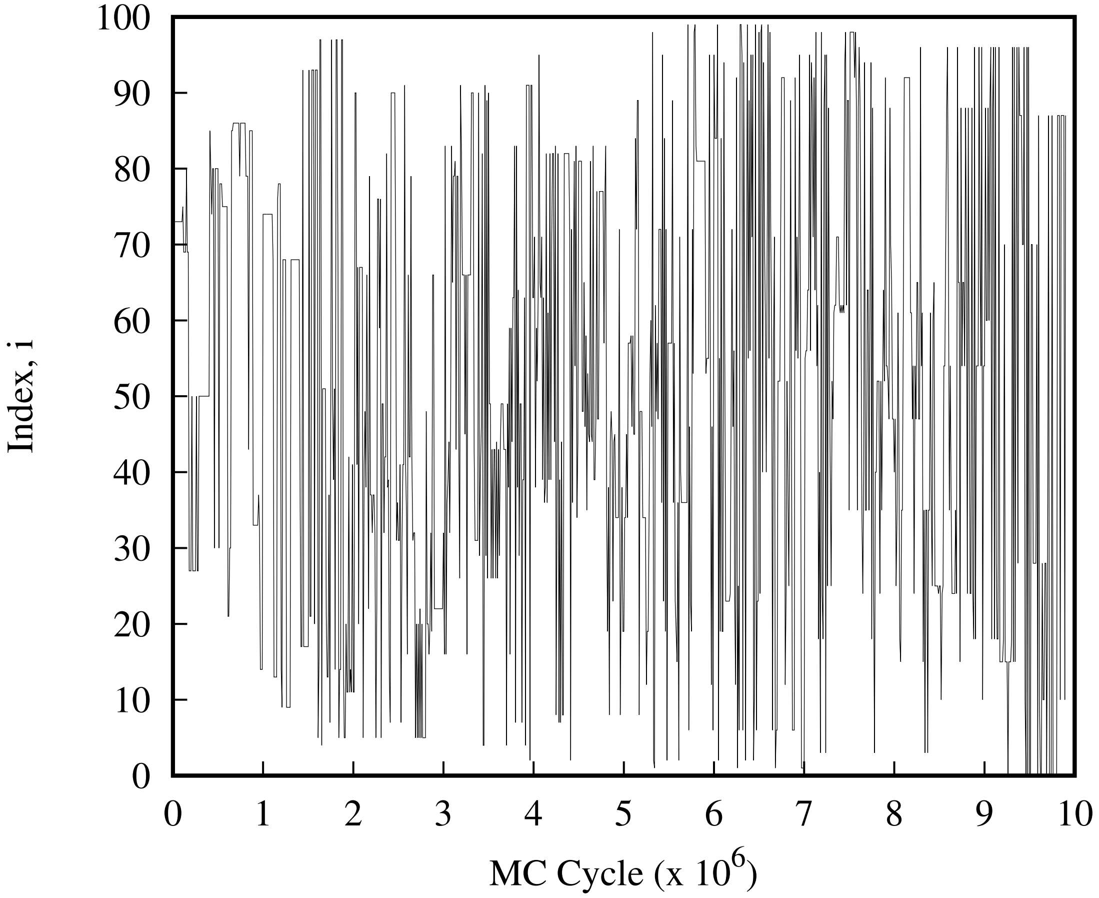

Figure 3.18: The change in parameter index over the course of the multiparameter simulation that is sampling the highest relative entropy states more frequently. Every 100,000 MC cycles, the most sampled index is removed, and the system is kicked from that state. The x-axis is in millions of Monte Carlo cycles, as denoted in the label.

After the first iteration, the final parameter set remaining became the center of a Gaussian distribution, and another $N = 100$ parameter sets were generated, and the process was repeated. The third and fourth iterations reduced the range in parameter space by letting $\sigma = 0.1$, and the final parameter set from the fourth iteration was determined to be the optimal parameter set, $\overline{\lambda}_{\text{opt}}$.

A fixed temperature simulation was run for the optimal parameter set, $\overline{\lambda}_{\text{opt}}$, and the RMSD distribution was calculated as before. Figure 3.19 shows the distribution of the RMSD (black) and the P(RMSD) for the MD simulation target ensemble, $\overline{\lambda}_{\text{MD target}}$, for reference.

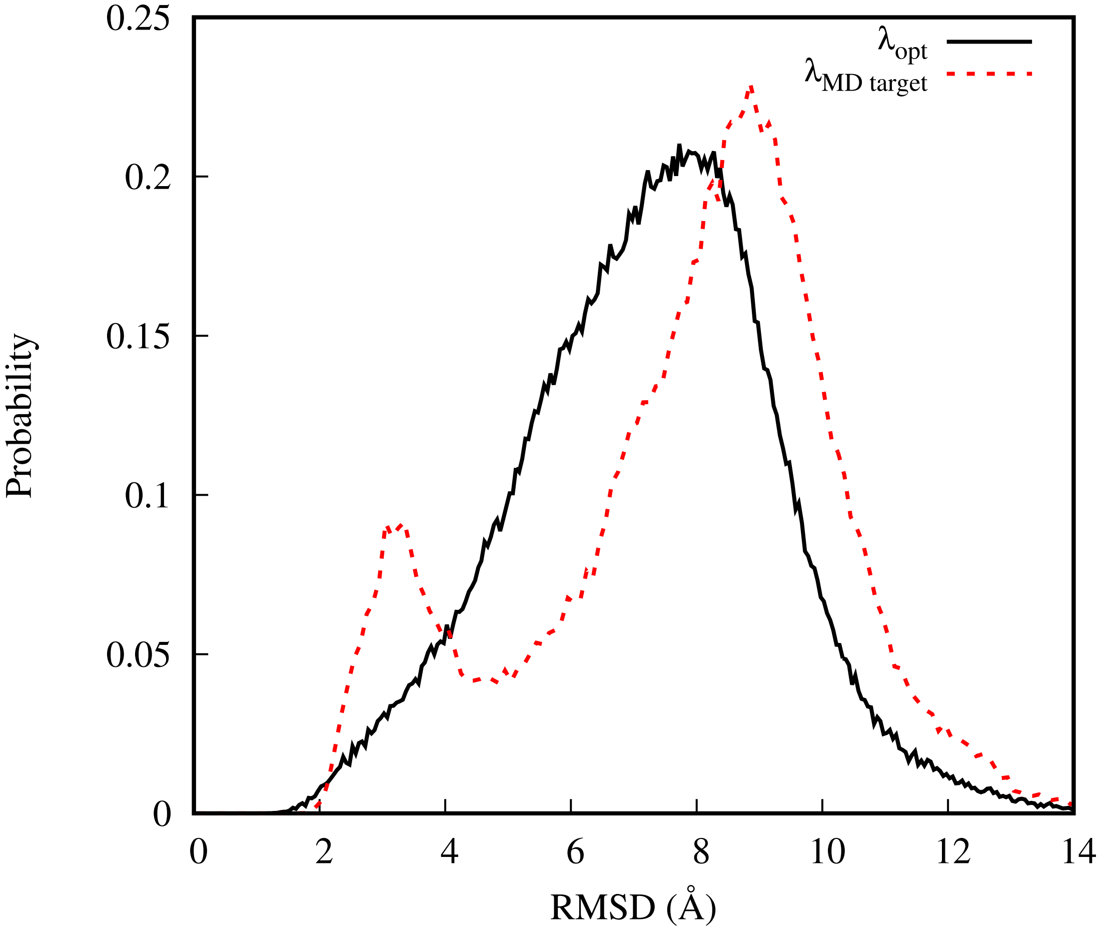

Figure 3.19: The RMSD histogram for the optimal parameter set, λ_{opt}, found after 4 iterations of the global optimization method. The RMSD distribution from the target MD simulation ensemble is plotted in red for reference.

The MD ensemble target RMSD was added to highlight the bimodal shape expected for two distinct regions, folded and unfolded. Although the optimal parameter RMSD distribution does not show the bimodal shape, it does show a shift in the RMSD to lower values. Furthermore, there is a number of configurations that fold into a structure similar to the native structure, RMSD $\leq$ 4Å.

3.4.3 Conclusion

The results shown above that use a global optimization method for determining the optimal parameter set are simple results used as a proof of concept that global optimization techniques can be used in addition to line minimization methods. The example provided here is one of the many different methods that could be used to carry out global searches in parameter space. Global optimization techniques are important in the case of a rough energy landscape that contained multiple minima, as they provide a way to escape local mimima regardless of energy barriers. Furthermore, the simple results here hint at the possibility of finding a more efficient method to systematically determine optimal CG model parameters than what was proposed in Chapter 2.

Chapter 4

Summary and Outlook

Coarse-grained computer models have become a powerful tool in studying certain properties of physical systems using computer simulations. The CG models give a simplified representation of the physical system of interest, and the degree of simplification depends on what physical properties are being studied. Many different CG models have been developed to study a large range of physical systems, and a variety of different ways to design the CG models have been developed. One of the main challenges in developing CG models is determining a potential energy function, E_{CG}, that allows the model to match the physical system. Often the potential energy function includes a number of unknown parameters, denoted $\overline{\lambda}$, which must be determined. The current methods to determine these unknown parameters, such as potential of mean force or multi-scale coarse-graining, rely on quantities that are computationally difficult to calculate.

Here, we developed a method to systematically determine the optimal model parameters for a CG model by minimizing the relative entropy, $S_{\mathrm{rel}}(\overline{\lambda})$. The relative entropy gives a statistical measure of the difference between two probability distributions, and for a molecular system, it is used to compare a CG ensemble of states and some target ensemble. The relative entropy depends on the difference in the CG and target potential energies, $\langle U_{\mathrm{CG}} - U_{\mathrm{T}} \rangle$, as well as the difference in the free energies, $F_{\mathrm{CG}} - F_{\mathrm{T}}$. A novel simulation method was developed which was based on a generalized ensemble Metropolis Monte Carla algorithm, and was used to directly determine the coarse-grained free energies, $F_{\mathrm{CG}}(\overline{\lambda})$. The systematic method minimized the relative entropy between the target ensemble and a CG ensemble generated for a set of CG model parameters.

The systematic method was applied to an existing coarse-grained model for protein folding. This was done by modifying the potential energy function of the CG model to include either 13 or 91 unknown model parameters. Then, the method systematically found the optimal parameter set for both cases by minimizing the relative entropy with respect to two different target ensembles: the single experimentally determined native structure [28], and an ensemble of configurations from an all-atom molecular dynamics simulation by the D. E. Shaw group [29]. It was shown that the optimal parameter set obtained for the potential energy function with 91 unknown parameters with the native structure as a target gave the best results. In this case, the majority of the configurations were in a folded state that closely matched the target native structure. Further results showed that the optimal parameter set found for the 91 parameter potential energy function gave better results than the 13 parameter set. Additionally, the all-atom ensemble target gave a poor optimal parameter set that did not fold into a native structure. This could be due to the fact that the all-atom ensemble was dominated by configurations in an unfolded state.

All in all, the systematic method for determining optimal CG model parameter sets was shown to be successful in determining the optimal parameters for a given set of unknown CG parameters. However, the degree to which the method determines an optimal parameter set greatly depends on the choice of the potential energy function and number of unknown parameters. Here, it was found that a more complex potential energy function with more model parameters was better able to capture the properties of the physical system. Furthermore, the validity of the optimal parameter set also depends on the choice of the target ensemble, and it was shown that to achieve proper folding of the protein sequence, the single experimentally determined native structure was the best choice for a target when minimizing the relative entropy.

Future work on this project could be to compare the optimal parameter sets presented in the results above to other protein sequences to see if the parameters are sequence independent and allow other sequences to fold to their individual native states. This would be of interest for developing a general CG protein folding model that would allow many given sequences to fold properly. If the optimal parameters found above are not general to any sequence, the systematic CG method could be modified to incorporate multiple target ensembles when minimizing the relative entropy. This modification might lead the method to determine the optimal model parameter set that could work for any protein sequence. This could be accomplished by running the method in parallel such that separate multiparameter simulations could be run on different sequences but with the same parameter set, and the relative entropy could be minimized and the information about the minima from all of the runs could be used to determine the next parameter set.

Another idea for future work is to extend the test on global optimization schemes, and determine the most efficient algorithm is for a global parameter space search. Global optimization would be very useful in the case where the energy landscape is not smooth, but instead contains many minima. The line minimization techniques applied in this project can be used any time the energy landscape is not excessively rough, since the generalized ensemble approach to the simulation allows for uniform sampling of states as long as the energy barrier is not large. However, if the landscape contained multiple minima and they were far apart in parameter space, or had high energy barriers, the line minimization techniques used here could get stuck in a local minima and not be able to escape.

Lastly, it is very important to highlight the fact that the systematic coarse-graining method for optimizing model parameters developed here is completely general to any coarse-grained model, with any potential energy function and unknown parameter set. Although the method was applied to a protein folding CG model, it could also be used to determine the optimal model parameters for CG models of other molecular systems, or even generally, any coarse-grained representations of any statistical-mechanical system.

Bibliography

[1] S. C. Glotzer *et al. International assessment of research and development in simulation-based engineering and science.* World Technology Evaluation Center, Baltimore, 2009.

[2] S. Izvekov and G. A. Voth. A multiscale coarse-graining method for biomolecular systems. *J. Phys. Chem. B*, 109(7):2469–2473, 2005.

[3] E. Brini, E. A. Algear, P. Ganguly, C. Li, F. Rodríguez-Ropero, and N. F. A van der Vegt. Systematic coarse-graining methods for soft matter simulations a review. *Soft Matter*, 9:2108–2119, 2013.

[4] S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid, and A. Kolinski. Coarse-grained protein models and their applications. *Chem. Rev.*, 116(14):7898–7936, 2016.

[5] N. Gō and H. Taketomi. Studies on protein folding, unfolding and fluctuations by computer simulation. iii. effect of short-range interactions. *Int. J. Pept. Protein Res.*, 13:235–252, 1979.

[6] O. Fornes, J. Garcia-Garcia, J. Bonet, and B. Oliva. On the use of knowledge-based potentials for the evaluation of models of protein-protein, protein-dna, and protein-rna interactions. *Adv. Protein Chem. Struct. Biol.*, 94:77–120, 2014.

[7] F. Ercolessi and J. B. Adams. Interatomic potentials from first-principles calculations: The force-matching method. *Europhys. Lett.*, 26(8):583–588, 1994.

[8] A. Lyubartsev and A. Laaksonen. Calculation of effective interaction potentials from radial distribution functions: A reverse monte carlo approach. *Phys. Rev. E*, 52(4):3730–3737, 1995.

[9] D. Reith, M. Pütz, and F. Müller-Plathe. Deriving effective mesoscale potentials from atomistic simulations. *J. Comput. Chem.*, 24:16241636, 2003.

[10] A. Chaimovich and M. S. Shell. Coarse-graining errors and numerical optimization using a relative entropy framework. *J. Chem. Phys*, 134:094112, 2011.

[11] M. S. Shell. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.*, 129(14):144108, 2008.

[12] S. Y. Mashayak, M. N. Jochum, K. Koschke, N. R. Aluru, V. Rühle, and C. Junghans. Relative entropy and optimization-driven coarse-graining methods in votca. *PLoS one*, 10, 2015.

[13] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley, New Jersey, USA, 2nd edition, 2006.

[14] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[15] U. H. E. Hansmann and Y. Okamoto. Generalized-ensemble monte carlo method for systems with rough energy landscape. *Phys. Rev. E*, 56(2):2228–2233, 1997.

[16] Y. Okamoto. Generalized-ensemble algorithms: enhanced sampling techniques for monte carlo and molecular dynamics simulations. *J. Mol. Graph. Model.*, 22(5):425–439, 2004.

[17] E. Marinari and G. Parisi. Simulated tempering: A new monte carlo scheme. *Europhys. Lett.*, 19(6):451–458, 1992.

[18] W. Nadler and U. H. E. Hansmann. Generalized ensemble and tempering simulations: A unified view. *Phys. Rev. E*, 75:026109, 2007.

[19] G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *J. of Comput. Phys.*, 23(2):187–199, 1977.

[20] M. R. Shirts and V.S. Pande. Comparison of efficiency and bias of free energies computed by exponential averaging, the bennett acceptance ratio, and thermodynamic integration. *J. Chem. Phys.*, 122:144107, 2005.

[21] C. H. Bennett. Efficient estimation of free energy differences from monte carlo data. *J. Comput. Phys.*, 22(2):245–268, 1976.

[22] A. M. Ferrenberg and R. H. Swendsen. Optimized monte carlo data analysis. *Phys. Rev. Lett.*, 63(12):1195–1198, 1989.

[23] M. R. Shirts and J. D. Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.*, 129:124105, 2008.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, New York, USA, 2nd edition, 1992.

[25] A. Bhattacherjee and S. Wallin. Coupled folding-binding in a hydrophobic/polar protein model: Impact of synergistic folding and disordered flanks. *Biophys. J.*, 102(3):569–578, 2012.

[26] S. P. Carmichael and M. S. Shell. A new multiscale algorithm and its application to coarse-grained peptide models for self-assembly. *J. Phys. Chem. B*, 116:8383–8393, 2012.

[27] C. Chipot and A. Pohorille, editors. *Free Energy Calculations: Theory and Applications in Chemistry and Biology.* Springer, Berlin, Germany, 2007.

[28] C. A. Sarisky and S. L. Mayo. The beta-beta-alpha fold: explorations in sequence space. *J.Mol.Biol.*, 307(5):1411–1418, 2001.

[29] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw. How fast-folding proteins fold. *Science*, 334:517–520, 2011.

[30] P. M. Pardalos, D. Shalloway, and G. Xue. Optimization methods for computing global minima of nonconvex potential energy functions. *J. Global Opt.*, 4:117–133, 1994.

[31] G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29:610–611, 1958.

[32] Z. Tan and J. Am. On a likelihood approach for monte carlo integration. *J. Stat. Phys.*, 99(468):1027–1036, 2004.

Appendix A

Derivation of MBAR Free Energy

The difference in the dimensionless free energies is given by

$$\Delta f_{ij} = f_j - f_i = -\ln \frac{c_i}{c_j} \tag{A.1}$$

where the ratio of the normalization constants must be solved to determine the estimating equations for the dimensionless free energies.

The method uses the identity

$$c_i \langle \alpha_{ij} q_i \rangle_i = c_j \langle \alpha_{ij} q_i \rangle_j \tag{A.2}$$

which holds for arbitrary functions α_{ij}, with the condition that c_i is non-zero, and the empirical estimator for the expectation value of g,

$$\langle g \rangle_i = N_i^{-1} \sum_{n=1}^{N_i} g(\boldsymbol{x}_i). \tag{A.3}$$

Combining these two equations and summing over the index j gives

$$\sum_{j=1}^{K} \frac{\hat{c}_i}{N_i} \sum_{n=1}^{N_i} \alpha_{ij} q_j(\boldsymbol{x}_{in}) = \sum_{j=1}^{K} \frac{\hat{c}_j}{N_j} \sum_{n=1}^{N_j} \alpha_{ij} q_i(\boldsymbol{x}_{jn}) \tag{A.4}$$

where $i = 1, 2, \dots, K$. Estimates of c_i from all of the sampled data are obtained from the solution to the set of equations for all $\hat{c}_i$. Note, Eq. A.4 is known as extended bridge sampling [32], as it gives a set of estimators that depend on the choice of function α_{ij}.

A choice for α_{ij} can be made such that the estimator obtained from A.4 is one that has been proven to be optimal (has the lowest variance for a large set of choices for α_{ij}). This choice is given by

$$\alpha_{ij}(\boldsymbol{x}) = N_j \hat{c}_j^{-1} \Bigg/ \sum_{k=1}^{K} N_k \hat{c}_k^{-1} q_k(\boldsymbol{x}). \tag{A.5}$$

Combining Eq. A.4 and Eq. A.5, gives

$$\sum_{j=1}^{K} \frac{\hat{c}_i}{N_i} \sum_{n=1}^{N_i} \frac{N_j \hat{c}_j^{-1} q_j(\boldsymbol{x}_{in})}{\sum_{k=1}^{K} N_k \hat{c}_k^{-1} q_k(\boldsymbol{x}_{jn})} = \sum_{j=1}^{K} \frac{\hat{c}_j}{N_j} \sum_{n=1}^{N_j} \frac{N_j \hat{c}_j^{-1} q_i(\boldsymbol{x}_{jn})}{\sum_{k=1}^{K} N_k \hat{c}_k^{-1} q_k(\boldsymbol{x}_{jn})} \tag{A.6}$$

where the left hand side can be rearranged and simplified. First, note that there is no summation over i, and the only term that depends on the summation over j is the numerator of the n summation. Therefore, the equation becomes

$$\frac{\hat{c}_i}{N_i} \sum_{n=1}^{N_i} \frac{\sum_{j=1}^{K} N_j \hat{c}_j^{-1} q_j(\boldsymbol{x}_{in})}{\sum_{k=1}^{K} N_k \hat{c}_k^{-1} q_k(\boldsymbol{x}_{in})} = \frac{\hat{c}_i}{N_i} \sum_{n=1}^{N_i} 1 = \hat{c}_i. \tag{A.7}$$

The right hand side simplifies to

$$\sum_{j=1}^{K}\sum_{n=1}^{N_j}\frac{q_i(\boldsymbol{x}_{jn})}{\sum_{k=1}^{K}N_k\hat{c}_k^{-1}q_k(\boldsymbol{x}_{jn})} \tag{A.8}$$

where the $\hat{c}_j/N_j$ terms cancelled.

Noting that the normalization constants can be written in terms of the dimensionless free energies as $f_i = -\ln c_i$, or $c_i = e^{f_i}$, Eq. A.6 becomes

$$\hat{f}_i = -\ln\sum_{j=1}^{K}\sum_{n=1}^{N_j}\frac{q_i(\boldsymbol{x}_{jn})}{\sum_{k=1}^{K}N_k e^{\hat{f}_k}q_k(\boldsymbol{x})}. \tag{A.9}$$

Replacing the unnormalized density function with the Boltzmann distribution, $q_i(\boldsymbol{x}) = e^{-U_i(\boldsymbol{x})}$, the equation for the estimated free energies is

$$\hat{f}_i = -\ln\sum_{j=1}^{K}\sum_{n=1}^{N_j}\frac{e^{U_i(\boldsymbol{x}_{jn})}}{\sum_{k=1}^{K}N_k e^{\hat{f}_k-U_k(\boldsymbol{x}_{jn})}} \tag{A.10}$$

as given in Eq. 1.14.

Appendix B

Gradient of the Relative Entropy

The gradient of the relative entropy with respect to the set of all CG parameters, λ_i, was found by Carmichael and Shell to be [26]

$$\frac{\partial S_{rel}}{\partial \lambda} = \beta \left\langle \frac{\partial U_{CG}}{\partial \lambda} \right\rangle_{AA} - \beta \left\langle \frac{\partial U_{CG}}{\partial \lambda} \right\rangle_{CG} \tag{B.1}$$

where the derivative of the CG potential energy with respect to the set of parameters is averaged over the CG ensemble and the AA ensemble separately.

Let the CG potential energy, which depends on the configuration, r, and the parameter λ, have the form

$$E_{CG}(r, \lambda) = E_0(r) + \lambda \varepsilon(r) \tag{B.2}$$

where $E_0(r)$ and $\varepsilon(r)$ are energy terms that are independent of the parameter, λ. If the parameter is dynamic, then a subscript j is used to denote the current index.

Using the above form for the energy and taking the derivative, the second term in Eq. 3.10 becomes

$$\beta\left\langle \frac{\partial U_{CG}}{\partial \lambda} \right\rangle_{CG} = \beta Z_{\lambda}^{-1} \sum_{r} \varepsilon(r) e^{-\beta E_{CG}(r,\lambda)} \tag{B.3}$$

where the canonical ensemble is used, with $Z = \sum_r e^{-\beta E(r)}$.

Note, the probability of the system being in configuration r with the dynamic parameter λ_j, is $P(r, \lambda_j) = Z_{\lambda_j}^{-1} \sum_r e^{-\beta E_{CG}(r,\lambda_j)}$. Multiplying and dividing by the probability gives

$$\beta\left\langle \frac{\partial U_{CG}}{\partial \lambda} \right\rangle_{CG} = \beta \frac{Z_{\lambda_j}}{Z_{\lambda}} \frac{1}{Z_{\lambda_j}} \sum_{r} \varepsilon(r) e^{-\beta E_{CG}(r,\lambda_j)} e^{+\beta\left(E_{CG}(r,\lambda_j) - E_{CG}(r,\lambda)\right)}. \tag{B.4}$$

Now, defining w as

$$w = e^{\beta\left(E_{CG}(r,\lambda_j) - E_{CG}(r,\lambda)\right)} = e^{\beta \Delta E} \tag{B.5}$$

and by the canonical definition of the partition function and the ensemble average, the ratio of the partition functions becomes

$$\frac{Z_{\lambda}}{Z_{\lambda_j}} = \frac{1}{Z_{\lambda_j}} \sum_{r} e^{-\beta E_{CG}(r,\lambda_j)} w = \langle w \rangle_{\lambda_j}. \tag{B.6}$$

Now, taking Eq. B.5, B.6 and the energy with the form given in Eq. B.2 and substituting into Eq. B.4, we obtain an equation for the gradient of the relative entropy,

$$\frac{\partial S_{rel}}{\partial \lambda} = \beta\left\langle \frac{\partial U_{\text{CG}}}{\partial \lambda} \right\rangle_{\text{AA}} - \beta \frac{\left\langle \varepsilon(r) w \right\rangle_{\text{CG},\lambda_j}}{\langle w \rangle_{\text{CG},\lambda_j}}. \tag{B.7}$$

To use this equation practically, the ensemble average is rewritten as a sum from 1 to M_j, which is the number of configurations for the i-th parameter value. So,

$$\langle e_{hp} w \rangle_{\lambda_j} = \frac{1}{M_j} \sum_{i=1}^{M_j} e_{hp}(r_i) e^{\beta(\lambda_j - \lambda) e_{hp}(r_i)} \tag{B.8}$$

and

$$\langle w \rangle_{\lambda_j} = \frac{1}{M_j} \sum_{i=1}^{M_j} e^{\beta(\lambda_j - \lambda) e_{hp}(r_i)} \tag{B.9}$$