Statistical Computation for Problems in Dynamic Systems and Protein Folding

Thesis supervisor

S.C. Samuel Kou

Author

Samuel Wing Kwong Wong

Statistical Computation for Problems in Dynamic Systems and Protein Folding

Abstract

Inference for dynamic systems and conformational sampling for protein folding are two problems motivated by applied data, which pose computational challenges from a statistical perspective. Dynamic systems are often described by a set of coupled differential equations, and methods of parametric estimation for these models from noisy data can require repeatedly solving the equations numerically. Many of these models also lead to rough likelihood surfaces, which makes sampling difficult. We introduce a method for Bayesian inference on these models, using a multiple chain framework that exploits the underlying mathematical structure and interpolates the posterior to improve efficiency. In protein folding, a large conformational space must be searched for low energy states, where any energy function constructed on the states is at best approximate. We propose a method for sampling fragment conformations that accounts for geometric and energetic constraints, and explore ideas for folding entire proteins that account for uncertain energy landscapes and learning from data more effectively. These ingredients are combined into a framework for tackling the problem of generating improvements to protein structure predictions.

Contents

Acknowledgements

My advisor, Samuel Kou, has contributed to the success of my graduate studies in many ways. During these years, he has provided me with guidance to sharpen my statistical thinking, while also giving me the freedom to develop my skills as a researcher. He had an important role in advising my work in all three chapters of this dissertation. In times when research was difficult and trying, he was as patient and encouraging as possible. He also supported me with funding for research and attending conferences, as well as computing resources for my projects. I have learned so much through working with Sam, and I am extremely grateful for all that he has done.

I thank my committee members, Jun Liu and Joe Blitzstein, for their help in refining this dissertation. Jun provided many suggestions in the development of the work presented in Chapters 2 and 3, and he was happy to give me advice and feedback throughout the protein folding project. Joe also deserves my thanks for helping to sharpen my teaching and research skills during my graduate studies.

I have worked in collaboration with other professors and students, whom I would like to acknowledge. I thank Jinfeng Zhang and Kevin Bartz, who have spent numerous hours on protein folding long before I joined the project. Their ideas and hard work led to an earlier version of the FRESS concept, and laid much of the foundation on which Chapter 3 is developed. Jinfeng was a helpful resource for the protein folding project as whole, and he was always available to assist and answer my questions. Some of the ideas presented in Chapter 2 were a result of joint work and discussions with Valeria Espinosa. I thank Xiao (Thomas) Tong, who collaborated with me for much of the work presented in Chapter 1. Jim Zidek deserves my thanks for advising

an ongoing research project on lumber properties, and for continuing to be a great mentor during my graduate school years.

On a more personal note, I would like to thank my fellow students in the Statistics Department for the friendships we've shared in this journey. I am also thankful for my friends in the Harvard Graduate Christian Community and Boston Chinese Evangelical Church. They have blessed me with their time, companionship, and hospitality during my time in Boston. And more importantly, they have always reminded me of the things that truly matter in life, especially in the most trying of times.

Finally, my parents have given me their constant love and care, even as I navigated the ups and downs of graduate school. I owe them my utmost gratitude.

To my father and mother, Peter and Sandra Wong.

Chapter 1

A Multiple-Chain Framework for Dynamic Systems Inference

We propose an efficient MCMC scheme for estimating parameters in dynamic systems governed by a set of ordinary differential equations (ODEs), which are frequently used to describe behaviors in science. The data observed are usually noisy and collected at discrete time intervals as the system evolves. Bayesian and likelihood-based inference for these systems face two main computational challenges: the rough shapes of likelihood surfaces encountered, and the time required for numerically solving the differential equations. We address the first of these challenges by proposing a framework that introduces a latent variable to control the noise level in the model, producing multiple chains of Monte Carlo samples of parameters to allow the coarser chains to improve convergence of the finer chains. Samples from the chains can be combined to provide more efficient estimates for quantities of interest. While this improves sampling for the rough posterior surfaces often encountered in these mod-

els, it still relies heavily on numerical ODE solvers, such as Runge-Kutta methods. Calling the numerical solver at every iteration creates a computational bottleneck, especially for larger models or stiff ODE systems. To tackle this second challenge and reduce the frequency at which the numerical solver is used, we propose the use of an interpolating function on the closeness of the solution to observed values, while retaining estimation accuracy.

1.1 Two examples

The parameter estimation problem for dynamic systems is motivated by the frequent use of coupled ordinary differential equations to describe behaviors in science. We begin by providing two concrete examples.

1.1.1 Oscillation of mRNA and protein levels in cultured cells

The oscillation of *hes1* mRNA and Hes1 protein levels in cultured cells, which exhibit the behavior of regulation via negative feedback, is described in Hirata et al. (2002): "a simple negative feedback loop, in which Hes1 represses transcription from the hes1 promoter, would be insufficient to maintain a stable oscillation, because this system would rapidly fall into equilibrium". The authors thus postulate the existence of a Hes1-interacting factor to explain this phenomenon. This system can be described

by the following set of nonlinear ordinary differential equations:

$$\begin{aligned} \frac{dx_1}{dt} &= -Ax_1x_3 + Bx_2 - Cx_1 \\ \frac{dx_2}{dt} &= Dx_2 + \frac{E}{1+x_1^2} \\ \frac{dx_3}{dt} &= -Ax_1x_3 + \frac{F}{1+x_1^2} - Gx_3, \end{aligned} \tag{1.1}$$

where x_1 is the concentration of Hes1 protein, x_2 is the concentration of hes1 mRNA, and x_3 is the concentration of the Hes1-interacting factor. The system has seven parameters: A, B govern the rate of protein synthesis in the presence of the interacting factor, C, D, G are the rates of decomposition, and E, F are inhibition rates.

After serum treatment, mRNA and protein levels are measured every 30-45 minutes. To illustrate the overall oscillatory behavior, Figure 1.1 shows the plot of mRNA and protein levels simulated from this system, assuming a measurement interval of 30 minutes. In practice, the sequence of observations on x_1 and x_2 over the course of the cell culture will also be subject to measurement uncertainty.

As an example of a parameter estimation problem, suppose that the decomposition and inhibition rates are fixed, and A, B are unknown. Assuming independent additive Gaussian measurement noise with $\sigma = 0.1$, a likelihood can be defined on the parameters by computing the normal densities at the observed data points, with means at the numerical solution corresponding to the parameters. Fixing $\sigma = 0.1$, a plot of this likelihood surface as a function of A and B appears in Figure 1.2. The shape is rough, with sharp ridges where the behavior of the ODE is very sensitive to parameter values.

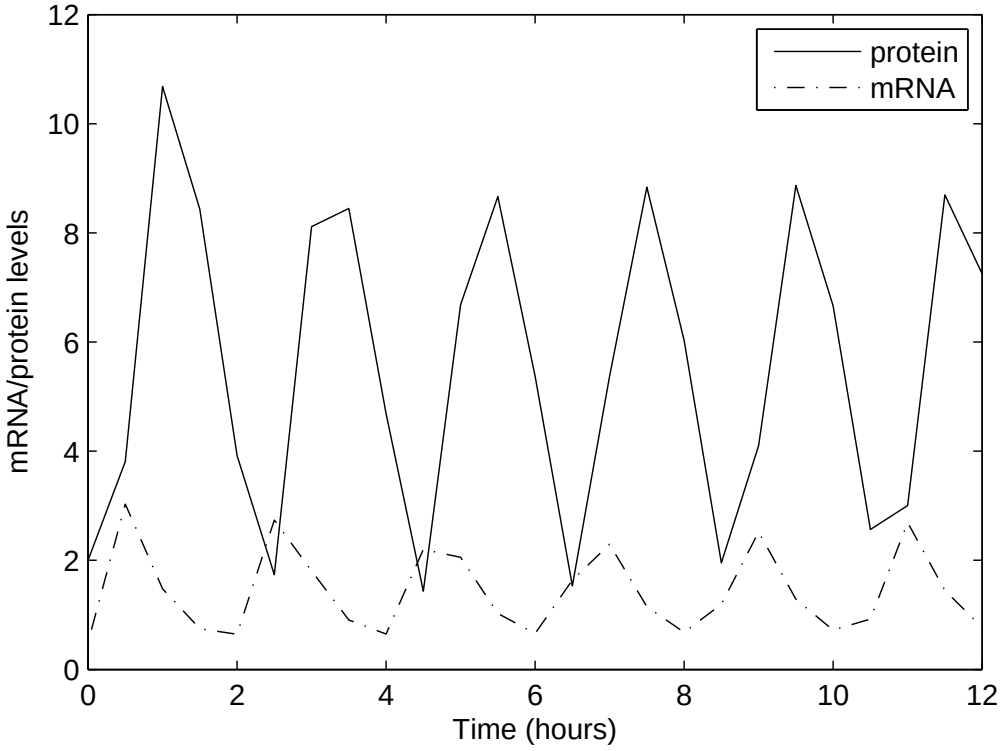


Figure 1.1: Data simulated from ODE system (1.1), with parameter values $A = 0.022$, $B = 0.3$, $C = 0.031$, $D = 0.028$; $E = 0.5$, $F = 20$, $G = 0.3$ as taken from Hirata et al. (2002). Plot shows hes1 mRNA and protein levels, assuming perfect measurements are taken every 30 minutes. An oscillatory cycle of approximately 2 hours can be observed.

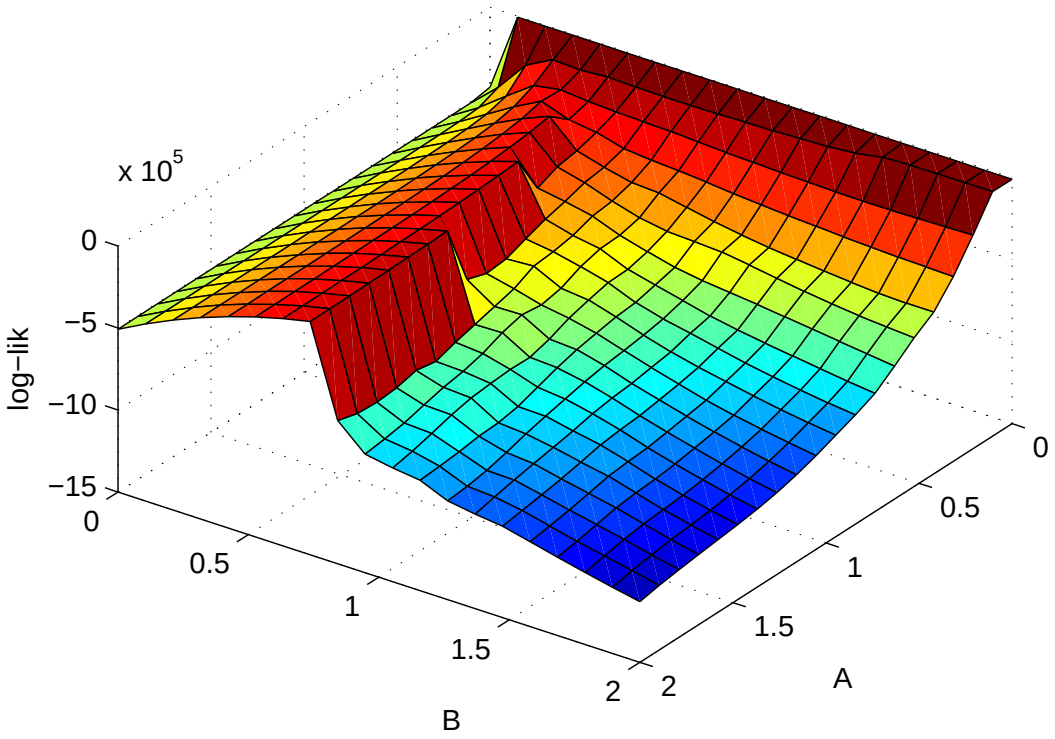


Figure 1.2: Likelihood surface of (1.1), when A, B are unknown. Additive Gaussian noise with $\sigma = 0.1$ has been added.

1.1.2 HIV viral fitness

Data from HIV viral fitness experiments in vitro are modeled by a set of ordinary differential equations by Miao et al. (2009). Replication fitness is evaluated on an assay, where different variants (e.g. mutant and wildtype) of HIV-1 viruses must compete for targeted cells in the same environment. A simplified model of cell counts can be expressed using the following set of ODEs (Miao et al., 2008):

$$\begin{aligned}
\frac{dT}{dt} &= (\rho - k_m T_m - k_w T_w - k_R T_{mw})T \\
\frac{dT_m}{dt} &= (k_m T - q_m T_w)T_m + 0.25 k_R T_{mw} T \\
\frac{dT_w}{dt} &= (k_w T - q_w T_m)T_w + 0.25 k_R T_{mw} T \\
\frac{dT_{mw}}{dt} &= 0.5 k_R T_{mw} T + (q_m + q_w) T_w T_m, \qquad (1.2)
\end{aligned}$$

where T, T_m, T_w, and T_{mw} are numbers of uninfected cells, cells infected by mutant virus, cells infected by wildtype virus, and cells infected by both; ρ is the net growth rate of T; (k_m, k_w, k_R) the infection rates of mutant virus, wildtype virus, and virus from dually infected cells, respectively; and q_m and q_w the dual infection rates. In this setting, approximate cell counts are observed at given time points; the unknown rates are the quantities of interest and must be estimated.

1.2 Background

The two examples illustrate the general setting under which estimates for parameters in dynamic systems are sought. We are given a set of ordinary differential

equations, often nonlinear,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, t|\theta)$$

where the vector $\mathbf{x}(t)$ is the list of N system outputs that evolve over time, over an interval $t \in [0, T]$, and θ is the vector of parameters that must be estimated from experimental data. Assuming that $\mathbf{f}$ is continuously differentiable with respect to $\mathbf{x}$, solutions of the ODE exist and are unique given initial values $\mathbf{x}(0)$. Most nonlinear ODE systems are not solvable analytically. Numerical methods (such as the fourth-order Runge-Kutta method) do provide reasonably accurate solutions, when parameter values are given.

In practice, experimental data from ODEs will be recorded at discrete time points, and may be noisy or subject to measurement error. In addition, some components of the system may not be observed (e.g. the hypothetical Hes1-interacting factor). Suppose that for each observable system component i, we obtain data $y_i(t_1), y_i(t_2), \ldots, y_i(t_n)$, for $0 < t_1 < t_2 < \ldots < t_n < T$, with

$$y_i(t_j) = g_i(x_i(t_j)) + e_{ij},$$

$i = 1, \ldots, N$ and $j = 1, \ldots, n$. Here, the noise term e_{ij} is assumed to be iid, additive and normal, after appropriate transformations g_i (if necessary) on the original measurements.

One of the earliest methods developed to estimate parameters under this setting is nonlinear least squares (NLS), as studied in Biegler et al. (1986). A trial set of parameters is chosen, and a numerical method (e.g. Euler discretization, Runge-

Kutta) is used to approximate the solution given the parameters and initial conditions, obtaining $\hat{x}_i(t_j|\Theta)$ for observed components $i = 1, \ldots, N$ and $j = 1, \ldots, n$. The parameter estimate is the vector $\hat{\theta}$ that minimizes the objective function

$$\sum_{i=1}^{N} \sum_{j=1}^{n} (y_i(t_j) - \hat{x}_i(t_j|\theta))^2$$

Methods such as gradient descent or Gauss-Newton can be used, for performing the minimization. The downsides of this approach are that a numerical solution of the ODE is required for each set of trial parameters, and additional computation is required to obtain interval estimates. Convergence might not be reliable if the starting point is poor.

More recently, a number of methods have been proposed to reduce the computational burden of repeatedly evaluating the ODE numerical solver. From a frequentist point of view, one class of such methods involves the construction of basis functions or splines. As a representative example, consider the generalized smoothing method developed by Ramsay et al. (2007), which eliminates the use of the numerical solver. The system components x_i are expressed in terms of a basis function expansion, where the number of basis functions is chosen so as to allow enough flexibility in the behavior of estimated functions $\hat{x}_i(t)$ to satisfy the ODE system. Much of the computational burden is thus shifted to obtaining estimates for both the parameters of interest θ as well as the coefficients of the basis functions. The method also provides linear approximations for interval estimates based on analytic derivatives.

In our work, we will take a Bayesian approach paired with Markov Chain Monte Carlo (MCMC) based methods. ODE models may suffer from identifiability issues,

as noted by Miao et al. (2009) and Huang et al. (2006). Practitioners may also have competing models in mind to describe the experimental data. Obtaining the entire posterior distributions of the parameters would be very useful for characterizing this type of uncertainty, which would not be as amenable from a likelihood-based perspective alone.

The simplest Bayesian approach is to apply a prior on the parameters θ, and allow the observed $\mathbf{y}$ at times t_j to follow

$$y_i(t_j)|\theta, \sigma^2 \sim N(\hat{x}_i(t_j|\theta), \sigma^2), \tag{1.3}$$

where $\hat{x}(t_j|\theta)$ denotes the numerical solution of the ODE system given the set of parameters, and error variance σ^2. The posterior distribution of θ is then given by

$$p(\theta|\mathbf{y}) \propto \pi(\theta) \prod_{i,j} p(y_i(t_j)|\theta, \sigma^2).$$

Standard Metropolis-Hastings techniques can then be used to update θ and draw samples from its posterior density. This idea was outlined in Gelman et al. (1996). This basic approach suffers from some of the same drawbacks as NLS, namely that many evaluations of the numerical solver are required, and convergence might be slow due to the rough posterior surface. A hierchical Bayesian extension of this approach was used by Huang et al. (2006) to study HIV dynamics; the model was fitted using standard Metropolis techniques with multivariate normal proposal densities. Due to the roughness of posteriors associated with dynamic models, tuning appropriate proposals can be difficult. It can be possible to reduce this problem by using adaptive

proposals, e.g. Haario et al. (2006), but convergence can still fail.

A general solution for sampling from rough, multimodal densities is parallel tempering, as introduced in Swendsen and Wang (1986). This has been applied for sampling from posteriors of ODE parameters, e.g. Campbell (2007). While this multiple-chain technique is generally applicable, sampling could potentially be more efficient if the error structure of the ODE model were to be exploited. Also for complicated posteriors, convergence may be faster when the histories of previous chains are saved, in the spirit of the equi-energy sampler (Kou et al., 2006). A development of this idea will be the focus of the remainder of this chapter.

1.3 A multiple-chain method

Our goal in this section is to provide a multiple-chain method to sample effectively from posteriors corresponding to ODE parameters, assuming normal measurement noise. The key ideas are as follows: (1) We will flatten out the likelihood by introducing artificial noise, at the level of measurement error. (2) Multiple Monte Carlo chains will be constructed, by controlling the artificial noise level. (3) The coarser chains will be used to speed up the convergence of finer chains.

Assume that measurement noise is iid and normal across all system components, and begin with the basic Bayesian formulation as in equation (1.3). Then, introduce a latent variable $\mathbf{z}(t_j)$, such that the following conditional distributions hold for the latent $\mathbf{z}(t_j)$ and the observed $\mathbf{y}(t_j)$:

$$\mathbf{y}(t_j)|\mathbf{z}(t_j), \theta, \sigma^2, \epsilon^2 \quad \sim \quad N(\mathbf{z}(t_j), \epsilon^2 I)$$

$$\mathbf{z}(t_j)|\theta, \sigma^2 \quad \sim \quad N(\hat{\mathbf{x}}(t_j|\theta), \sigma^2 I). \tag{1.4}$$

We are free to choose the noise parameter ϵ, which serves to flatten out the likelihood and thus also increases the ease of drawing samples and navigating the space. In other words, $\mathbf{z}(t_j)$ is an artificially noise-contaminated version of the ODE solution. At $\epsilon = 0$, we recover the original model.

The likelihood function for any particular choice of ϵ is

$$\begin{aligned} L(\theta, \sigma^2|\mathbf{Y}) &= \prod_{j=1}^{n} p(\mathbf{y}(t_j)|\theta, \sigma^2) \\ &= \prod_{j=1}^{n} \int p(\mathbf{y}(t_j)|\mathbf{z}(t_j), \theta, \sigma^2, \epsilon^2) p(\mathbf{z}(t_j)|\theta, \sigma^2) d\mathbf{z}(t_j). \end{aligned}$$

With normal errors, the integral can be computed analytically, directly giving

$$\mathbf{y}(t_j)|\theta, \sigma^2, \epsilon^2 \sim N\left(\hat{\mathbf{x}}(t_j|\theta), (\sigma^2 + \epsilon^2) I\right).$$

As before, specifying priors on θ, σ^2 completes the posterior density. The log-posterior for an artificial noise term ϵ can then be written as

$$\begin{aligned} &\log p(\theta, \sigma^2|\mathbf{Y}, \epsilon) = \\ &-\frac{nN}{2} \log(\sigma^2 + \epsilon^2) - \frac{1}{2(\sigma^2 + \epsilon^2)} \sum_{i=1}^{N} \sum_{j=1}^{n} (y_i(t_j) - \hat{x}_i(t_j|\theta))^2 + \log \pi(\theta, \sigma^2) + \text{const.} \end{aligned}$$

The primary effect of ϵ is to rescale the sum of squares discrepancy (i.e. the objective function of NLS) from the observed data.

Sampling with this framework begins with a first chain, where we choose a value ϵ

sufficiently large, such that a tuned Metropolis-Hastings algorithm (e.g. with multivariate normal proposals) can adequately explore the surface of the noise-contaminated model. Initial parameter values are chosen, and an appropriate burn-in period is run. Samples during the burn-in are discarded; subsequent samples are collected to form an empirical distribution for this chain.

Sampling for the second and subsequent chains proceeds as follows. First, pick a value of ϵ smaller than the previous chain, and randomly draw a starting set of parameters from the previous chain. Then, at each parameter update step, with probability $1 - p$ a regular MCMC step is run. With probability p, a set of parameters is uniformly drawn from the previous chain, which is used as an independent Metropolis proposal. One limitation of parallel tempering is that swaps between chains can only occur between current states. Instead, we draw from the entire posterior distribution of the previous chain. The proposals drawn from the higher level, coarser chain facilitate faster convergence for the chain at hand. A burn-in period is run as before, after which samples again are collected for an empirical distribution. This procedure continues until the final chain, where the noise term is set at $\epsilon = 0$ to recover the original model.

The sampling scheme can be formalized as follows.

A multiple-chain scheme for sampling ODE parameter posteriors

Let $p^{(i)}(\theta) \equiv p(\theta, \sigma^2|\mathbf{Y}, \epsilon_i)$, where the sequence of ϵ_i satisfy $\epsilon_1 > \epsilon_2 > \ldots > \epsilon_K = 0$

Choose an initial value $\theta_0^{(1)}$.

For $m = 1, 2, \ldots$

perform a MH step to update $\theta_{m-1}^{(1)}$ to $\theta_m^{(1)}$ as a draw from $p^{(1)}(\theta)$

if $m >$ burn-in

save $\theta_m^{(1)}$ as sample for construction of empirical distribution $\hat{p}^{(1)}(\theta)$

For $i = 2, \ldots, K$

draw $\theta_0^{(i)}$ uniformly from $\hat{p}^{(i-1)}(\theta)$

For $m = 1, 2, \ldots$

with probability $1 - p$, perform a MH step to update

$\theta_{m-1}^{(i)}$ to $\theta_m^{(i)}$ as a draw from $p^{(i)}(\theta)$

with probability p, draw a proposal θ^* uniformly from $\hat{p}^{(i-1)}(\theta)$

let $\theta_m^{(i)} = \theta^*$ with probability $\min\left(1, \dfrac{p^{(i)}(\theta^*)p^{(i-1)}(\theta_m^{(i)})}{p^{(i)}(\theta_m^{(i)})p^{(i-1)}(\theta^*)}\right)$

let $\theta_m^{(i)} = \theta_{m-1}^{(i)}$ otherwise

if $m >$ burn-in

save $\theta_m^{(i)}$ as sample for construction of empirical distribution $\hat{p}^{(i)}(\theta)$

1.4 Numerical illustrations

Consider fitting the mRNA/protein oscillator model, with data generated as described above. First assume that all three components are observed. Set vague priors Gamma$(0.001, 0.001)$ for A and B, and IGamma$(0.001, 0.001)$ for σ^2. Set $p = 0.3$, the probability of drawing from the previous chain at any given step. For the Metropolis updates at each iteration, A, B are sampled together from the posterior and σ^2 is

sampled separately. In the first chain, we begin sampling from a poor starting value, $A = B = \sigma = 1$. A fourth-order Runge-Kutta numerical solver is used for each set of sampled parameters.

Trace plots and autocorrelations for the multiple-chain scheme are shown in Figure 1.3. Starting with a sufficiently large artificial noise, the first chain has adequate convergence properties with a vanilla MH sampler. As we move to subsequent chains, a noticeable improvement in the autocorrelation plots can be seen, owing to the independent proposals drawn from the previous chain. The true values $A = 0.022$ and $B = 0.3$ are generally well-covered by the samples over the different noise levels, and the mode becomes sharper as ϵ is reduced. Since the roles of ϵ and the measurement error σ overlap, its distribution shifts the most as ϵ is reduced to zero. The true value of $\sigma^2 = 0.01$ only becomes apparent in the final chain. See Figure 1.4.

The application of this sampling scheme only requires a small adjustment when system components are partially observed. In this mRNA and protein Hes1 example, the Hes-1 interacting factor is hypothetical and cannot be observed. The corresponding terms in the likelihood are integrated out, but otherwise estimation proceeds in a similar manner. The smoothed density estimates are compared for the final chain in Figure 1.5. It can be seen that the posterior distributions obtained using the complete data are sharper and with less bias, than with a missing component.

Next, consider the HIV model. One replicate of the experimental data, as reported in Miao et al. (2009) appears in Table 1.1. With this real data example, there is no true parameter value; the objective is to provide the best estimates for the given model structure. It seems reasonable to assume in this case that the measurement errors are

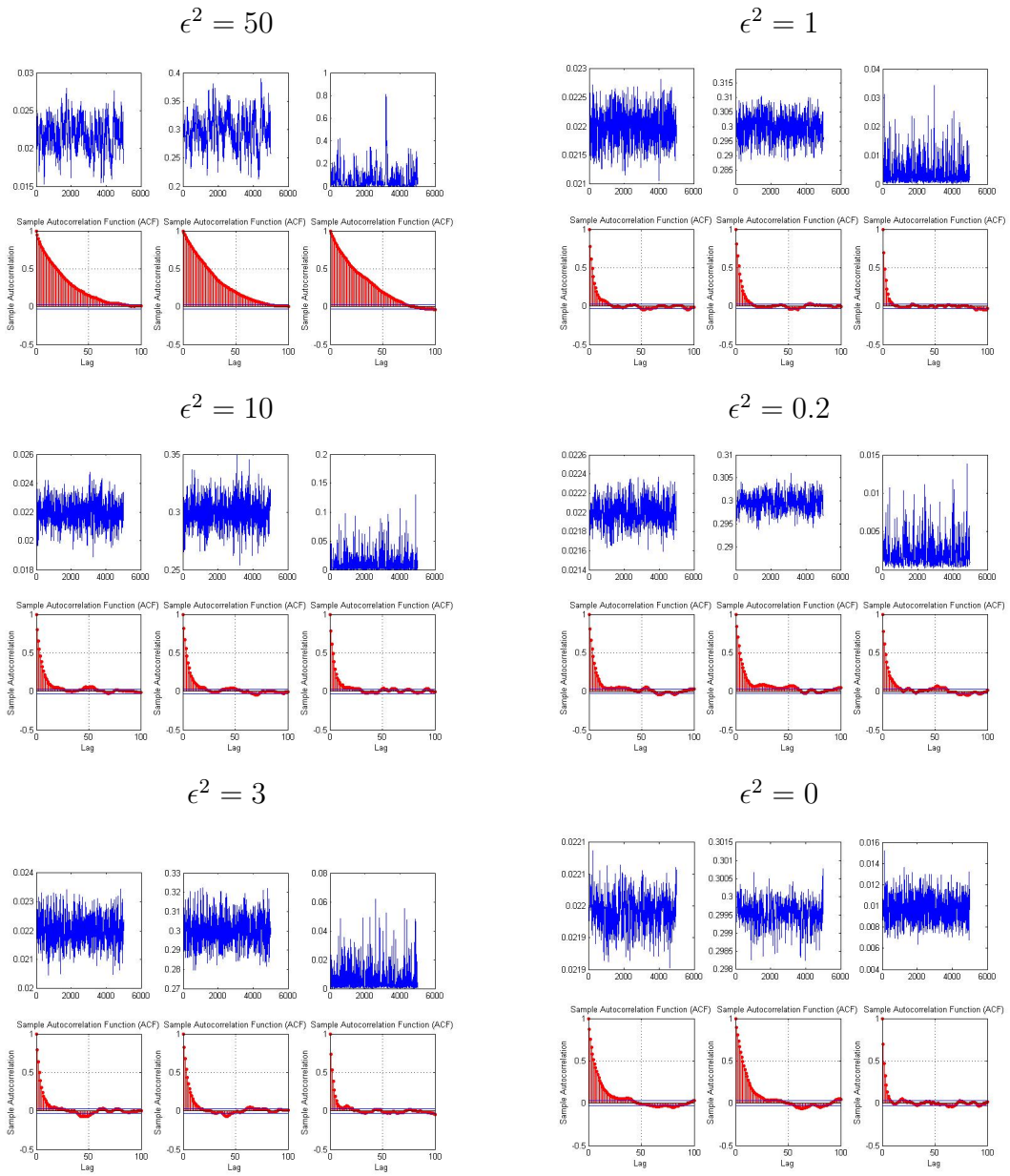


Figure 1.3: Trace plots and autocorrelations from multiple-chain scheme for mRNA and protein oscillatory example. The sequence of artificial noise levels are indicated in the figure.

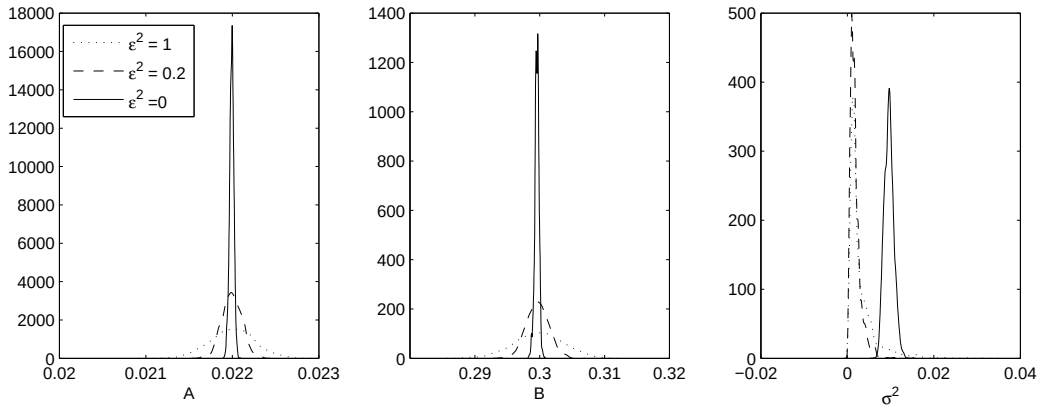


Figure 1.4: Kernel density estimates, illustrating the shifts in the posterior density of the parameters over changes in the noise level ϵ.

iid, normal, and additive after a log transformation, i.e. $\log(y_i(t_j)) = \log(\hat{x}_i(t_j)) + e_{ij}$. The results for posterior parameters are listed in Table 1.2. We find that our 95% intervals are somewhat narrower than those reported by the authors for this particular model, and the point estimates are similar.

Table 1.1: Measured numbers of HIV infected cell counts, taken from Miao et al. (2009).

time (hours)	T	T_m	T_w	T_{mw}
70	32,554,830	134,173	26,180	9,818
94	46,645,200	481,950	103,950	18,900
115	64,240,540	1,230,460	309,260	26,320
139	65,563,680	9,863,280	3,000,480	1,364,580
163	36,366,400	36,545,600	10,281,600	28,806,400

1.5 Implementation heuristics

Here we provide some practical guidelines for implementing the proposed sampling scheme.

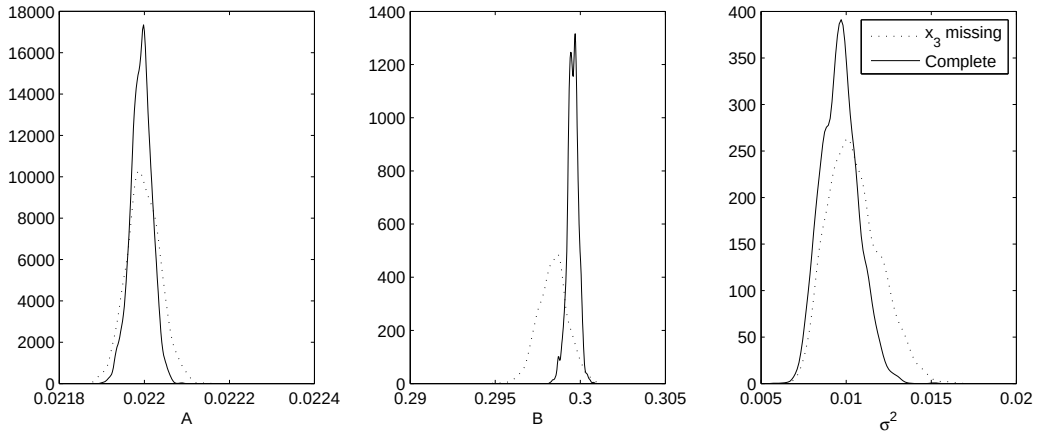


Figure 1.5: Kernel density estimates for final chain, comparing posteriors of fully observed system (solid) with mRNA and protein levels only (dashed).

Table 1.2: Parameter estimate and posterior intervals for HIV data.

Parameters	Estimate(median)	Central 95% posterior interval	
ρ	1.41E-02	1.26E-02	1.55E-02
k_m	1.16E-09	1.14E-09	1.19E-09
k_w	1.30E-09	1.28E-09	1.36E-09
k_R	5.06E-10	4.86E-10	5.77E-10
q_m	3.62E-09	3.59E-09	3.65E-09
q_w	1.56E-09	1.49E-09	1.58E-09
σ^2	9.85E-02	8.61E-02	1.15E-01

For the first chain, the choice of ϵ is very important. It is essential that the first chain is able to explore the entire space; if a region is missed in the first chain, subsequent chains will be very unlikely to sample those regions, as the posterior surface becomes increasingly rough. The reliability of subsequent chains will be only as good as the chains before it. Therefore, if the nature of the posterior is not well known, it is advisable to set ϵ as large as necessary. For this purpose it may be helpful to run the numerical solver for a number of selected points within the likely parameter space, to gauge the range of the log-posterior. Setting ϵ_1 to be quite large may increase the number of chains K ultimately required to bridge the model to $\epsilon_K = 0$, but this tradeoff may be necessary to obtain correct results. Also, note that there is no upper limit on this allowable artificial noise level relative to the true measurement error; the posterior only becomes further flattened.

We also recommend tuning the proposal variance (assuming MVN proposals) during each chain. A short number of iterations can be run, and adjustments can be made adaptively. Earlier chains should have larger proposal step sizes, since the posterior surface is flatter. The purpose of the the MH move in later chains is for local exploration, and intuitively will use smaller proposal step sizes. These chains will rely on the cross-chain proposal to move to distinct regions in the parameter space.

The ladder of ϵ for subsequent chains should be chosen to maintain a reasonable acceptance rate for the cross-chain move, when it is proposed. Again, there is a tradeoff. If the ratio of $\epsilon_i/\epsilon_{i-1}$ is too small, the distributions $p^{(i)}(\theta)$ and $p^{(i-1)}(\theta)$ can be quite different. In this case, $p^{(i-1)}(\theta)$ is not a very good proposal density for $p^{(i)}(\theta)$ and most samples will be rejected. This defeats the purpose of constructing

the previous chain to improve convergence of the next chain. On the other hand, if $\epsilon_i/\epsilon_{i-1}$ is too close to 1, then $p^{(i)}(\theta)$ and $p^{(i-1)}(\theta)$ will overlap significantly. Most samples will be accepted, but such a choice does little to bridge the gap towards the ultimate $\epsilon = 0$ chain. As such, more chains than necessary would be constructed, losing efficiency. In practice, we find that a cross-chain acceptance rate of around 30% is a reasonable compromise.

Another adjustable setting is the frequency of attempting cross-chain moves. Since our cross-chain sample bears resemblance to the drawing of samples from empirical energy rings in the equi-energy sampler (Kou et al., 2006), that provides a basic guideline. Heuristically, we suggest a similar or somewhat higher frequency of these attempts compared to the equi-energy sampler, in the range of 30%, as we are drawing from the entire distribution. This setting can be dynamically tuned, if necessary, based on the observed autocorrelations as the chain is sampled.

Finally, we note that the scheme can be implemented either on a single CPU, or parallelized over multiple CPUs. If running on a single CPU, subsequent chains are run in order. The coarser chain is stopped when sufficient samples of $\hat{p}^{(i)}(\theta)$ have been collected, and the next chain is started. Sampling then proceeds to the next chain. When multiple CPUs are available, coarser chains do not have to be stopped. We can continue to build samples for $\hat{p}^{(i)}(\theta)$ after the next chain is started. The pool of available draws for $\hat{p}^{(i+1)}(\theta)$ continues to grow in this case, providing a more accurate empirical distribution to be drawn from by the next chain. The extra samples available from higher chains can also be useful for the final inference, as shown in the next section.

1.6 Inference with multiple chains

In this section, suppose we are interested in inference for some function of the parameter vector $h(\theta)$, such as tail probabilities and posterior intervals. In the implementation of the multiple chain scheme, the final chain with $\epsilon_K = 0$ corresponds to the original model, for which we have draws from the distribution $p^{(K)}(\theta)$. The quantity of interest is then

$$E_K[h(\theta)] = \int h(\theta) p^{(K)}(\theta)\, d\theta.$$

The simplest unbiased estimate is the sample mean based on the empirical distribution $\hat{p}^{(K)}(\theta)$ which contains the draws $\theta_1^{(K)}, \ldots, \theta_m^{(K)}$,

$$\hat{E}_K^{(K)}[h(\theta)] = \frac{1}{m} \sum_{j=1}^{m} h(\theta_j^{(K)}). \tag{1.5}$$

The sampler also provides draws from the distributions of the previous chains $p^{(1)}(\theta), \ldots, p^{(K-1)}(\theta)$, which are based on varying levels of added artificial noise ϵ. The question of interest is how to use the samples saved in $\hat{p}^{(1)}(\theta), \ldots, \hat{p}^{(K-1)}(\theta)$ to improve the estimation of $E_K[h(\theta)]$. We follow the reasoning outlined in Kong (1992). For each chain $i = 1, \ldots, K-1$, we will require an importance weight adjustment. Let $w^{(i)}(\theta) = p^{(K)}(\theta)/p^{(i)}(\theta)$. Then an unbiased estimate based on the i-th chain is

$$\hat{E}_K^{(i)}[h(\theta)] = \frac{\sum_{j=1}^{m} h(\theta_j^{(i)}) w^{(i)}(\theta_j^{(i)})}{\sum_{j=1}^{m} w^{(i)}(\theta_j^{(i)})}$$

This is equivalent to the usual importance weight-adjusted estimate, $\hat{E}_K^{(i)}[h(\theta)] =$

$\sum_{j=1}^{m} h(\theta_j^{(i)})w^{(i)}(\theta_j^{(i)})$, when the weights have been standardized, i.e. $\tilde{w}^{(i)}(\theta_j^{(i)}) = w^{(i)}(\theta_j^{(i)})/\bar{w}^{(i)}$, where $\bar{w}^{(i)}$ denotes the sample average of the weights for chain i. This procedure yields K unbiased estimates of $E_K[h(\theta)]$, which must be combined into a single estimate. One approach is to weight them inversely proportional to their variances, disregarding the dependence.

Write $v^{(i)} = h(\theta^{(i)})w^{(i)}(\theta^{(i)})$, then the above estimate can be re-expressed as $\hat{E}_K^{(i)}[h(\theta)] = \bar{v}^{(i)}/\bar{w}^{(i)}$. Letting $g(v,w) = \frac{v}{w}$, with derivatives $g_v(v,w) = \frac{1}{w}$ and $g_w(v,w) = -\frac{v}{w^2}$, applying the Delta method gives

$$\begin{aligned} Var^{(i)}\begin{pmatrix}\bar{v}^{(i)}\\ \bar{w}^{(i)}\end{pmatrix} &\approx \frac{1}{m}\begin{pmatrix}-\frac{E(V^{(i)})}{E(W^{(i)})^2} & \frac{1}{E(W^{(i)})^2}\end{pmatrix}\begin{pmatrix}\sigma_{VV} & \sigma_{VW}\\ \sigma_{VW} & \sigma_{WW}\end{pmatrix}\begin{pmatrix}-\frac{E(V^{(i)})}{E(W^{(i)})^2}\\ \frac{1}{E(W^{(i)})^2}\end{pmatrix} \\ &= \frac{1}{m}\left(\sigma_{VV}E(V^{(i)})^2 + \sigma_{WW} - \sigma_{VW}2E(V^{(i)})\right), \end{aligned}$$

since $E(W^{(i)}) = 1$, and where the expectations and covariances are computed over $p^{(i)}(\theta)$. Sample quantities can thus be substituted to complete the approximation. The final estimate is then

$$\hat{E}_K[h(\theta)] = \frac{\sum_{i=1}^{K}[\widehat{Var}^{(i)}(\bar{v}^{(i)}/\bar{w}^{(i)})]^{-1}\hat{E}_K^{(i)}[h(\theta)]}{\sum_{i=1}^{K}[\widehat{Var}^{(i)}(\bar{v}^{(i)}/\bar{w}^{(i)})]^{-1}}. \tag{1.6}$$

The gain in efficiency by combining the chains can be measured by comparing the mean-squared error (MSE) of the estimates. The estimated MSE is computed by using the Monte Carlo samples to evaluate the function h, and consider the ratio of MSEs when using all chains (1.6) to using the final chain only (1.5), that is, $\widehat{MSE}_{all}/\widehat{MSE}_K$. As an illustration, consider estimating the one-sided posterior probability $P(A >$

0.022) in the oscillator model. For this low-dimensional example, the true probability can be computed by a numerical integration. We find $\widehat{MSE}_{all}/\widehat{MSE}_K = 0.57$.

1.7 Interpolating the posterior

One drawback of the scheme is the continued reliance on the ODE numerical solver at each iteration to produce the posterior value for sampled parameters. This step becomes increasingly slow for more complicated models where many samples are required for MCMC convergence, and for stiff ODE systems where adaptive step-size numerical solvers face slow convergence issues of their own. The next goal is to reduce the reliance on the numerical solver, and we pursue an interpolation approach.

The idea is to reuse numerical solutions that have been computed for Monte Carlo samples. By looking at the form of the log-posterior of the i-th chain $\log p^{(i)}(\theta)$, the term that depends on the numerical solver is $\frac{1}{2(\sigma^2+\epsilon^2)} \sum_{i=1}^{N} \sum_{j=1}^{n} (y_i(t_j) - \hat{x}_i(t_j|\theta))^2$. Let

$$SS(\theta) \equiv \sum_{i=1}^{K} \sum_{j=1}^{n} (y_i(t_j) - \hat{x}_i(t_j|\theta))^2,$$

which is not dependent on the values of σ^2 or ϵ^2. As the sampler runs for a longer period of time, the value of $SS(\theta)$ can be stored each time a numerical solution is computed, with the goal of covering a wide range of parameter values. Under our multiple chain setup, this formulation is very useful since the SS values are reusable across all of the chains.

As more samples accumulate, for a proposal θ^*, we may be able to approximate the posterior value by interpolating its value based on nearby samples of θ that had

numerical solutions computed. This step must be much faster than the numerical solution. Thus, while splines or basis functions might be used for this purpose, a simple weighted average of nearby points is appealing. We consider a linear interpolant of the form

$$\widehat{SS}(\theta^*) = \frac{\sum_j w_j SS(\theta_j)}{\sum_j w_j},$$

where the sum runs over a set of θ_j in the neighborhood of θ^*.

1.7.1 Choosing the interpolation neighborhood

The choice of the set of θ_j is important for the goodness of the interpolation. In spatial applications, a simple and appealing choice is inverse distance weighting (IDW), e.g. Lu and Wong (2008). IDW is sensible when it can be assumed that the contributions of sampled points are inversely related to distance, for the value being interpolated. It is however unlikely that this assumption will be generally valid in the ODE parameter space, since the behavior of the system can change rapidly (e.g. the likelihood plot of the mRNA and protein example). Samples that are somewhat nearby, but not near enough to capture more rapid changes in $SS(\theta)$, will not contribute useful information if used in the interpolation. Restricting consideration to a very local set of θ_j would be preferred. The key criterion then becomes the choice of which specific samples to include.

One promising approach is motivated by computational mathematics. A *triangulation* of a bounded subset of N-dimensional Euclidean space decomposes the set into N-simplices, such that the intersection of any two simplices in the decomposition has dimension less than N, or is empty (Chen and Xu, 2004). This is an extension of

the usual use of the term triangulations, from planes in $\Re^2$ to arbitrary dimension. A commonly-used triangulation, the *Delaunay triangulation*, as applied to a finite set of points V, satisfies the property that no vertices in V are inside the circumsphere of any simplex in the triangulation.

The use of the Delaunay triangulation for functional interpolation purposes has been studied by Omohundro (1989); the proposed approach to high-dimensional interpolation is to compute a triangulation the input points and use linear interpolation within each simplex. This leads to a continuous, piece-wise linear interpolated function within the convex hull of the input points. This is a convenient property for our ODE estimation context, since only the samples that form the vertices of the bounding simplex will be given weights in the interpolation.

We briefly mention some of its optimality properties, from an interpolation perspective. Omohundro (1989) shows that if the second derivative is bounded, the maximum error possible with the Delaunay triangulation is less than with any other triangulation. Chen and Xu (2004) show that an optimal Delaunay triangulation, in the sense of minimizing interpolation error, exists for any given convex continuous function. While this optimal triangulation can be difficult to compute, a bounding Delaunay simplex suitable for the interpolation of a new sample point can be determined by local criterion in a computationally efficient manner (Omohundro, 1989).

1.7.2 Sampling scheme with interpolation

We now describe how interpolation of the posterior can be incorporated into the ODE parameter estimation problem. As the first sampling chain is started, numerical

solutions for sampled θ are computed, and the corresponding $SS(\theta)$ are stored. This continues until a reasonable set of samples have been obtained. Then, for subsequent proposals of new θ^*, we attempt compute the local bounding simplex and its corresponding interpolated value if triangulation is successful. Triangulation will fail if the θ^* is not within the convex hull of the collected samples; in this case, the numerical solution of θ^* should be computed, and $SS(\theta^*)$ added to the list.

If an interpolated value $\widehat{SS}(\theta^*)$ exists, a decision must be made to accept or reject it for use in the computation of the Metropolis ratio. This decision should depend on the estimated interpolation error and the degree to which the SS function is changing. To estimate this interpolation uncertainty, we take a cross-validation approach. Discard the N points used for the interpolated value, and compute a new simplex using the next set of nearest points, to give $\widehat{SS}(\theta^*)_{CV}$. The interpolation uncertainty translates into uncertainty in the resulting Metropolis ratio used to accept or reject θ^*. Intuitively, the allowable error in the Metropolis ratio must be small enough to avoid noticeable impacts on the stationary distribution. Thus, if $\widehat{SS}(\theta^*)_{CV} \approx \widehat{SS}(\theta^*)$, the corresponding approximate Metropolis ratio can be applied for the acceptance or rejection of θ^*. Otherwise, we compute the numerical solution for θ^* and add $SS(\theta^*)$ to our list. Note that relative to the computational cost of fully evaluating the numerical solution, the overhead of this extra step is low.

The possible error in the Metropolis ratio can be approximated via

$$\Delta MH \equiv \frac{1}{2(\hat{\sigma}^2 + \epsilon^2)} \left| \widehat{SS}(\theta^*) - \widehat{SS}(\theta^*)_{CV} \right|,$$

where the current Monte Carlo estimate of the value of the noise σ^2 is substituted,

and our heuristical guideline is to use the interpolation if $\Delta MH < 0.1$, and compute a new numerical solution otherwise.

As before, samples for the first chain are collected in this manner, providing the approximate empirical distribution $\hat{p}^{(1)}(\theta)$. Similarly, the next chain begins with a draw from $\hat{p}^{(1)}(\theta)$. Then, for the local M-H update, a list of values for $SS(\theta)$ is already available from the first chain. These values can immediately be used for interpolation as the second chain begins. While we expect some interpolations to be successful in this second chain by using these values, it is likely that further numerical solutions will be required. This is because the second chain has a rougher posterior surface with a smaller ϵ^2. Therefore, changes in $p^{(2)}(\theta)$ will be more pronounced for the same change in $SS(\theta)$. As a result, some regions of the parameter space will now require additional numerical solutions to satisfy the above heuristic. This is intuitively sensible, as we would like a finer and more precise estimate of the regions of the posterior that are close to the modes. The same logic applies to subsequent chains.

The sampling scheme with interpolation can be formalized as follows.

A multiple-chain scheme with interpolation

for sampling ODE parameter posteriors

Let $p^{(i)}(\theta) \equiv p(\theta, \sigma^2|\mathbf{Y}, \epsilon_i)$, where the sequence of ϵ_i satisfy $\epsilon_1 > \epsilon_2 > \ldots > \epsilon_K = 0$

Choose an initial value $\theta_0^{(1)}$.

For $m = 1, 2, \ldots$

propose θ_m^* as a draw from $p^{(1)}(\theta)$

if $m < M$, the number of initial numerical solutions to collect

compute the numerical solution for θ_m^* and store $SS(\theta_m^*)$

otherwise

compute the bounding simplex of θ_m^*

if bounding simplex does not exist

compute the numerical solution for θ_m^* and store $SS(\theta_m^*)$

else compute $\widehat{SS}(\theta_m^*)_{CV}$ and $\widehat{SS}(\theta_m^*)$

if $\Delta MH > 0.1$ compute the numerical solution for θ_m^* and store $SS(\theta_m^*)$

update $\theta_{m-1}^{(1)}$ to $\theta_m^{(1)}$ according to Metropolis ratio

(ratio based on $\widehat{SS}(\theta_m^*)$ if interpolation successful, else $SS(\theta_m^*)$)

if $m >$ burn-in

save $\theta_m^{(1)}$ as sample for construction of empirical distribution $\hat{p}^{(1)}(\theta)$

For $i = 2, \ldots, K$

draw $\theta_0^{(i)}$ uniformly from $\hat{p}^{(i-1)}(\theta)$

For $m = 1, 2, \ldots$

with probability $1 - p$, propose θ_m^* as a draw from $p^{(iS)}(\theta)$

compute the bounding simplex of θ_m^*

if bounding simplex does not exist

compute the numerical solution for θ_m^* and store $SS(\theta_m^*)$

else compute $\widehat{SS}(\theta_m^*)_{CV}$ and $\widehat{SS}(\theta_m^*)$

if $\Delta MH > 0.1$ compute the numerical solution and store $SS(\theta_m^*)$

update $\theta_{m-1}^{(i)}$ to $\theta_m^{(i)}$ according to Metropolis ratio

with probability p, draw a proposal θ^* uniformly from $\hat{p}^{(i-1)}(\theta)$

let $\theta_m^{(i)} = \theta^*$ with probability $\min\left(1, \frac{p^{(i)}(\theta^*)p^{(i-1)}(\theta_m^{(i)})}{p^{(i)}(\theta_m^{(i)})p^{(i-1)}(\theta^*)}\right)$

let $\theta_m^{(i)} = \theta_{m-1}^{(i)}$ otherwise

if $m >$ burn-in

save $\theta_m^{(i)}$ as sample for construction of empirical distribution $\hat{p}^{(i)}(\theta)$

1.7.3 Example

As an illustration of this methodology, again consider the simulated mRNA/protein data in section 1.1.1. We take $M = 50$, the number of initial numerical solutions to collect before attempting interpolation. As a benchmark, a normal Metropolis sampler with numerical solvers takes around 1260 sec for 10,000 iterations. We ran six chains with this scheme, and the results are summarized in Table 1.3. The overall fraction of ODE solutions required is in the range of $10-20\%$ for this data. Note that as anticipated, subsequent chains do require additional ODE solutions, to interpolate more finely in important regions of the posterior. The overall time savings is around 4-fold; the overhead of computing local interpolations is empirically justifiable.

It is important that the posterior draws from the interpolated distribution remain faithful to the original posterior. This is illustrated in Figure 1.7.3. The top panel shows the histograms of the posterior draws of the parameters A and B under the original and interpolation schemes; they are virtually indistinguishable. To visually see the effect of interpolation, the difference between the true posterior and interpolated posterior for the final $\epsilon = 0$ chain is shown in the bottom panel. We note

that most of the error occurs in regions that are further away from the parameter estimates. The interpolation is quite accurate in the regions of interest, as a result of the heuristic Metropolis ratio criterion.

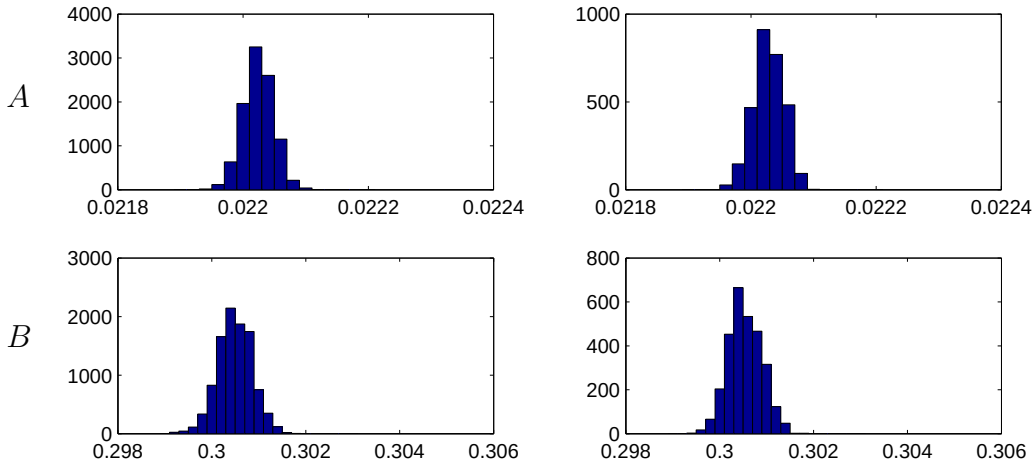
Table 1.3: Performance of multiple-chain scheme with interpolation, for the mRNA/protein levels model.

ϵ^2	Iterations	Number of ODE evaluations	Cumulative ODE evaluations	Time (sec)	Cross-move acceptance
50	10000	1297	1297	257.6	N/A
8	10000	1293	2590	261.4	0.269
1	10000	1546	4136	292.2	0.220
0.1	10000	1907	6043	328.3	0.174
0.02	10000	1286	7329	261.9	0.342
0	10000	1323	8652	267.3	0.575

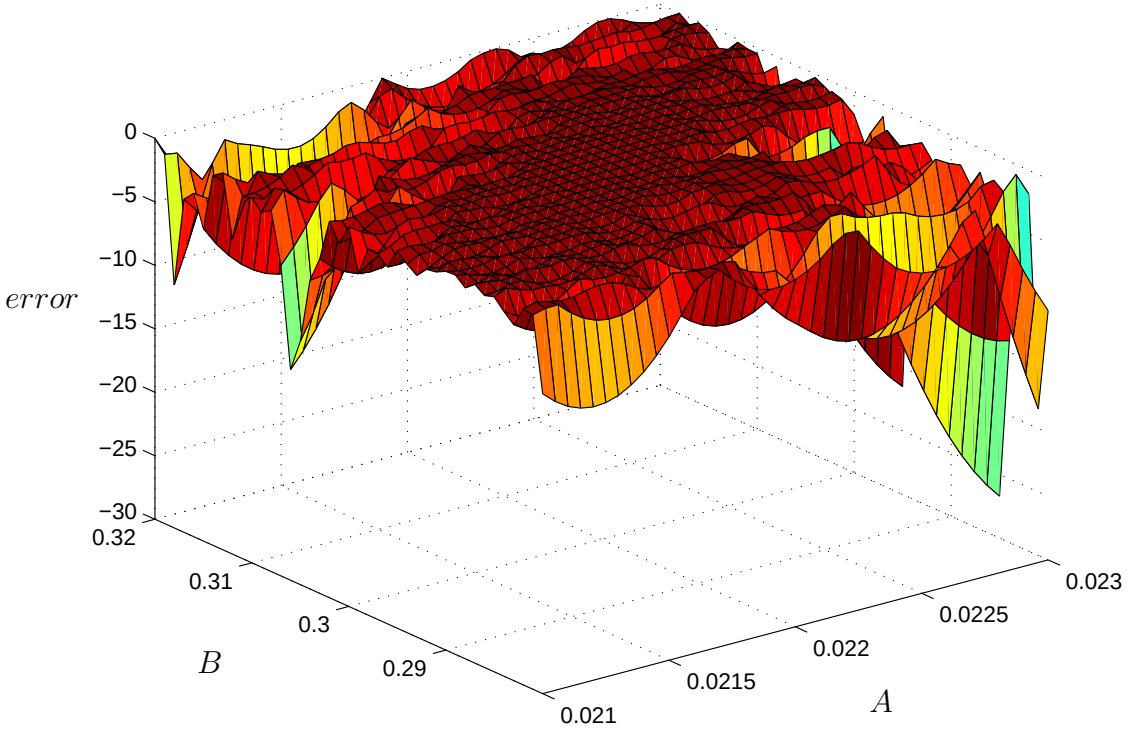
1.8 Conclusions and future directions

We have proposed a multiple-chain sampling scheme for performing Bayesian inference in parameter estimation problems of ODE models. The scheme assumes that observed data are noise-contaminated and recorded at discrete time points. A latent variable is introduced to control a level of artificial noise in the system, designed such that chains can borrow information from the previous chain to improve convergence. Only minor adaptations are required when certain system components are missing. The multiple-chain scheme helps to overcome the rough posteriors encountered in dynamic models, and can be more efficient sampler than an application of parallel tempering. Samples from chains can be combined to produce more efficient estimates (with lower MSE) for quantities of interest.

Next, we introduced interpolation methods to reduce the frequency at which nu-



(a) Comparison of parameter histograms
Left: interpolated scheme. Right: original scheme.



(b) Absolute error in interpolated $SS(A, B)$ surface.

Figure 1.6: Comparison of original and interpolated sampling schemes, for the mRNA/protein levels model.

merical solutions of the ODE are required. We used results from computational mathematics to design our approach. The method suggested works well with the multiple-chain setup, and naturally leads to finer interpolation in posterior regions of greater significance. We were able to maintain accurate inference by carefully selecting appropriate heuristics.

The work presented in this chapter present a number of possible directions for future research. In one direction, we might further reduce the use of the numerical solver by incorporating sensitivity analysis of ODEs, a concept common in applications of dynamic systems and the applied mathematics literature. Sensitivities of the ODE solution to perturbations in parameters are often computed to gauge the stability of the system. From a Monte Carlo sampling point of view, this information can be used in the context of evaluating the smoothness of the posterior and providing further guidance to sampling effort. In a second direction, we would like to scale up the method to work for much larger systems, such as encountered in systems biology. This direction would pose additional computational challenges to be tackled.

Chapter 2

A Statistical Framework for Protein Structure Refinement

The prediction of three-dimensional structure of proteins from their amino acid sequence has been of great interest in computational biology, since the discovery that sequence is generally sufficient for structure prediction (Anfinsen et al., 1973). The determination of structure by experimental methods has been attained for many proteins using the techniques of X-ray crystallography and nuclear magnetic resonance; at the same time this process is time-consuming, and not always successful as some proteins are either difficult to crystallize, or fail to be crystallized (Slabinski et al., 2007). In applications such as protein and enzyme design where a chosen three-dimensional structure is required, there are often too many candidate sequences for experimental determination to be feasible, and computational alternatives must be used (Kuhlman et al., 2003). There has thus been great interest in the bioinformatics community in the development of protein folding algorithms to predict structure from

sequence, e.g. see Kryshtafovych et al. (2005) and the groups mentioned therein. In this chapter, we describe statistical approaches to tackle some of the computational challenges arising from the protein folding problem, and construct a system to refine candidate structures.

2.1 Introduction to refinement

Given an amino acid sequence for a new protein of interest, construction of a structure prediction often begins with comparative (or homology) modeling (Martí-Renom et al., 2000). This is a procedure that aligns the new sequence with sequences of proteins of known structure (templates); on the basis of sequence similarity with templates, a three-dimensional structure is built. This procedure is based on the assumption that small changes in sequence will generally lead to only small changes in structure. The database of known structures that can be used as templates continues to grow as more proteins are determined experimentally, and these structures are available in the Protein Data Bank (PDB) (Berman et al., 2000). With the availability of more data, schemes for template detection have also grown in sophistication and power. One powerful method for template detection is HHPred (Söding et al., 2005), that finds remote homologies via a pairwise comparison of profile hidden Markov models. The resulting sequence alignment can be used as input for comparative modeling software such as MODELLER (Eswar et al., 2006), which builds a three dimensional structure from the input alignment.

These advances have consequently reduced the need to build structure predictions *ab initio* from the sequence, which was the original goal of packages such as ROSETTA

(Simons et al., 1999a). The goal of *ab initio* protein folding is to predict structure from sequence alone, without the aid of comparative modeling; this is most often useful when matching templates cannot be found. In these circumstances, the search for possible conformations is usually guided by an energy function and a sampling algorithm (Baker and Sali, 2001), with the goal of locating the global energy minimum for the sequence.

Both the construction of an energy function and the choice of sampling algorithm pose formidable computational challenges. The purpose of the energy function is to guide conformational sampling and optimization towards the truth, and the ideal energy landscape would be a "funnel" with the true conformation at the bottom of a deep energy well (Wolynes, 2005); it must also be able to distinguish misfolded structures from correct folds (Simons et al., 1999b). Many energy functions have been designed for use in protein folding that incorporate a combination of physics-based terms (such as Van der Waals forces) and statistically-learned terms, e.g. Simons et al. (1999a); Fujitsuka et al. (2004); Zhang et al. (2004); Shen and Sali (2006); Zhang and Zhang (2010); Liang et al. (2011); Zhou and Zhou (2002). These functions, while useful for various applications, still possess many inaccuracies when evaluated over sets of decoy structures, e.g. Tsai et al. (2003).

The search for a global energy optimum in protein folding, even in the case of lattice models, is NP-complete (Berger and Leighton, 1998); the space of conformations with a Cartesian coordinate representation for individual atoms is much larger. An efficient sampling algorithm is therefore crucial, and again many strategies have been proposed, e.g. Kim et al. (2009); Zhang et al. (2002); Bruccoleri and Karplus

(1990). Even with an accurate energy function, the true state of a protein may never be sampled (Kim et al., 2009).

The energy function and sampling algorithm continue to have an important role in the context of template-based modeling. Since the structure built from alignment is not expected to be perfect, a *refinement* step usually ensues, e.g. Roy et al. (2010), where the goal is to generate an improvement upon the template-based structure. Regions in the protein that lack template matches in particular will require additional sampling effort, e.g. Jacobson et al. (2004). Thus, while sampling and energy might not be applied as the only ingredients for the structure prediction process, they are definitely necessary for any refinement procedure on a given structural template. The refinement problem has been shown to be difficult in practice, and it is difficult to guarantee improvements in the structure (MacCallum et al., 2011).

Our goal is to develop approaches to the refinement problem that accounts for the aforementioned statistical challenges. In this chapter, we develop a selection procedure for predictions, and an overall scheme for sampling conformations based on Monte Carlo optimization. The sampling of segments within a protein that are not well modeled by matching templates is known as loop sampling (Lee et al., 2010), and will be the focus of the next chapter.

2.2 Constructions of energy functions

Most energy functions used for protein folding involve a linear combination of component terms. This idea originates in the work of Simons et al. (1999a); we briefly review this general approach. Ideally, an energy function should be useful for

guidance during Monte Carlo sampling of conformations, and the best prediction is the structure with the lowest energy. The authors identified a number of component energy terms that contribute to the distinguishing of good folds from misfolded structures, for any given amino acid sequence; these terms were later expanded and updated to provide the scoring function used in the ROSETTA package (Rohl et al., 2004a).

Each component term describes one aspect of a protein structure that may differ between native-like structures and misfolds. Some of the key components are: steric repulsion (VDW forces), torsion angle preferences, hydrogen bonding, solvation, atom-atom interactions, and side chain torsion angles. Van der Waals forces are physics-based and can be expressed with this functional form (based on the 12-6 Lennard Jones approximation),

$$E_{VDW} = \sum_{i>j} \left[\left(\frac{r_{ij}}{d_{ij}} \right)^{12} - 2 \left(\frac{r_{ij}}{d_{ij}} \right)^{6} \right] e_{ij}$$

where i, j are atom indices, and d_{ij} is the separation between the atoms. The sum runs over all atom pairs in the structure. The e_{ij} (geometric mean of atom well depths) and r_{ij} (VDW radii) have experimentally determined values. In contrast, many other components are often knowledge-based, that is, estimated from statistics in the PDB. For example, the term for main chain torsion angle preferences in the ROSETTA package is expressed as

$$E_{rama} = -\sum_{i} \log P(\phi_i, \psi_i | aa_i, ss_i), \tag{2.1}$$

where empirical densities for torsion angle pairs (ϕ_i, ψ_i), conditioned on amino acid type and secondary structure type, have been learned from the PDB, and the sum runs over all amino acids i in the sequence. For instance in these two terms, we expect that native-like structures would tend to have low VDW energies and have a profile of torsion angles that is realistic and similar to known proteins. These terms could thus help distinguish possibly viable structures from unrealistic ones. Similar logic applies to other component energy terms that have been developed, e.g. the popular DFIRE potential (Zhou and Zhou, 2002). Overall, the goal is that the complete energy function contains all the characteristics necessary for this classification task.

A question of interest then arises, when considering the manner in which information in the various energy components should be pooled together. The usual method is to sum them with linear weights. Simons et al. (1999a) describe the rationale as follows, to compute the probability density of a given structure based on a set of k energy components which we shall denote $E_1, \ldots, E_k$. Let E_{tot} be the total energy of the structure; then its density in the corresponding Boltzmann distribution (disregarding temperature) is $\exp(-E_{tot})$. The other energy terms can be expressed as a density in similar fashion. Suppose the terms yield independent density functions, then they can be multiplied to produce the density $\exp(-E_{tot})$,

$$\exp(-E_{tot}) = \prod_{i=1}^{k} \exp(-E_i),$$

or equivalently, $E_{tot} = \sum_{i=1}^{k} E_i$. The authors argue that the component densities provide some overlapping information to some extent, violating independence. Additionally, there can be overcounting or lack of independence even within an individual

term; for example, in the statistical torsion angle term (2.1), the (ϕ_i, ψ_i) of adjacent amino acids are not truly independent. To approach these issues, the authors propose instead the expression

$$\exp(-E_{tot}) = \prod_{i=1}^{k} [\exp(-E_i)]^{w_i},$$

a logarithmic pooling of the density functions, with weights w_i that must be estimated. This yields the linear form of composite energy functions usually seen, $E_{tot} = \sum_{i=1}^{k} w_i E_i$.

Estimation of the weights w_i can proceed in a number of ways. A commonly-used approach is the use of collections of "decoys". This can be used for weight-fitting as in Simons et al. (1999a), but also applicable for fitting parameters for energy functions in general, see for example Chuang et al. (2008); Liang et al. (2011). For a given protein with known structure, decoys are computationally generated incorrect folds of the same amino acid sequence. Since the space of possible incorrect structures is so large, good decoys should have favorable energy in at least some criteria, e.g. low Van der Waals forces indicating that no steric clash has occurred. To generate a set for training purposes, a number of proteins are usually chosen and decoys are generated for each (a few hundred to thousands of decoys, per protein). In the case of Simons et al. (1999a), approximately 30,000 decoys were generated for each of 21 proteins. Linear regressions were then fit with the k energy terms as predictors, and where the response variable is a proxy for how close a given decoy is to the native structure. The weights were then assigned to the estimated regression coefficients corresponding to each term.

Other types of objective functions based on decoys could be used. For example,

Liang et al. (2011) minimize an objective function of the form

$$\sum_{training} \frac{\sum_{decoys} \exp\{-E_{decoy}\}}{\exp\{-E_{native}\} + \sum_{decoys} \exp\{-E_{decoy}\}},$$

which accomplishes a similar purpose of training the energy function to separate natives from decoys.

Alternatively, energy parameters such as weights could be fitted by maximum likelihood on a set of native structures, as briefly mentioned by Simons et al. (1999a). Another approach that does not use decoys, instead defines a reference state relative to the native structure, as in DFIRE (Zhou and Zhou, 2002).

Ideally, constructed energy functions should have sufficient accuracy for use in both the sampling conformations, as well as ranking the goodness of structures when a list of decoys is provided. We next illustrate some of the deficiencies of energy functions and propose solutions.

2.3 Structure ranking

In this section, the goal is to select the correct native conformation among a collection of decoys. We use data from the biannual CASP experiment (Moult et al., 2009) for evaluation. It possesses desirable properties as a decoy set; the decoys are structure predictions generated by participating groups, where the groups are blinded. The predictions have native-like properties that can be difficult for a trained energy function to distinguish from the true structure, and thus poses a challenging test for an energy function to rank them correctly.

2.3.1 Construction of ranking function

For a fixed set of energy terms, we would like to develop a composite energy total that has a high probability of correctly identifying the native. We use structure predictions from the 8th CASP experiment and a standard decoy set in Tsai et al. (2003) as a training set. From the CASP8 set, we have 120 proteins with roughly 80 groups submitting predictions for each. The Tsai decoy set is built on 62 proteins, with 100 decoys each that score well using the default Rosetta energy terms, plus another 20 decoys that are randomly selected configurations. Structures from the 9th CASP experiments will be used to test the approach. The energy terms considered were the 18 components of the full-atom ROSETTA scoring function (Rohl et al., 2004a).

We note the intuition provided for the use of a linear weights on energy terms is based on uncertainty about the degree to which information in the terms overlap or are dependent. The logarithmic pooling of the densities is one possible ad-hoc method to combine them. For ranking purposes, this pooling function might be improved by relaxing the assumptions of linearity and the absence of interactions. Generalized additive models (GAMs), first introduced in Hastie and Tibshirani (1986) are a powerful tool that could be applied here, which we briefly review.

The main idea of using a GAM is to move away from a linear function, and model the dependence of a response Y on predictors $X_1, \ldots, X_k$ in a more non-parametric fashion. The usual GLM attempts to fit

$$g(E(Y)) = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + error,$$

for regression coefficients β and link function g, is extended is extended in the GAM context to

$$g(E(Y)) = s_0 + \sum_{i=1}^{k} s_i(X_i) + error,$$

where the s_i are unspecified smooth functions to be estimated. In the original implementation, the form of the s_i's was a simple scatterplot smoother; though as computational power has increased, the use of kernel or spline smoothers has become more popular. As would be expected, a greater amount of data required to fit a GAM adequately, due to the increase in degrees of freedom (Wood, 2006a). Fitting the s_i's typically requires the addition of some form of penalty to the likelihood to prevent overfitting, with the penalty based on a measure of degree of wiggliness in the smooth function.

In the original design of GAMs, there was no facility to handle the inclusion of interaction terms that were also non-parametric smooths. The fitting of multivariate smooth terms was later developed in Wood (2006b), which proposed a method to construct low-rank tensor product smooths, that could accommodate varying degrees of wiggliness in the univariate smooths. Suppose a construction of a bivariate smooth on covariates X_1, X_2 is required, and that their individual smooths can be written

$$s_1(x_1) = \sum_{i=1}^{n_1} \alpha_i a_i(x_1), \qquad s_2(x_2) = \sum_{j=1}^{n_2} \beta_i b_i(x_2),$$

for sets of basis functions $\{a_i\}$ and $\{b_j\}$ that are of low rank. Using the basis for x_2,

the α_i can be allowed to also vary smoothly with x_2, i.e.

$$\alpha_i(x_2) = \sum_{j=1}^{n_2} \beta_{ij} b_j(x_2),$$

which leads to a natural expression for the joint smooth, namely

$$s_{1,2}(x_1, x_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \beta_{ij} a_i(x_1) b_j(x_2).$$

Similarly, the penalties different degrees of wiggliness in the marginals are combined into a wiggliness measure for the joint smooth, weighted to allow the penalty to be invariant to covariate scaling.

In this context it is natural to consider, as an alternative to a linearly weighted total energy, a GAM of the form

$$E_{GAM} = \sum_{i=1}^{k} s_i(E_i) + \sum_{\text{some } i,j} s_{i,j}(E_i, E_j)$$

where for the pairwise terms, the most important interactions are selected. Application to the decoy training set requires a few considerations. First, a response variable must be defined. While in Simons et al. (1999a) the definitions used were cutoffs in root-mean-squared distance (RMSD) of the decoy from the native structure, its shortcomings as a metric for whole-protein similarity have been criticized in favor of methods that give more credit to local correctness, such as the Global Distance Test (GDT) (Zemla, 2003). GDT searches for the longest continuous segments within the protein that can be aligned with the true structure within a given RMSD cutoff. This

is the metric we employ here. Second, there should be a protein-specific adjustment, to account for the differences in the average quality of the generated decoys over the different proteins. Third, there must be sufficient data to prevent overfitting. While this aspect might not be critical for structure selection, overfitting could be detrimental if the function is used for sampling, as explored later. It is not difficult to generate more decoys for each protein; the challenge is to generate *representative* decoys that cover important regions of the vast conformational space.

Letting there be M distinct proteins, and N_M decoys for each, we fit the following model based on these considerations,

$$GDT_{m,n} = \sum_{i=1}^{k} s_i(e_{i,m,n}) + \sum_{\text{some } i,j} s_{i,j}(e_{i,m,n}, e_{j,m,n}) + \beta_m + \epsilon_{m,n}, \tag{2.2}$$

for $m = 1, \ldots, M$ and $n = 1, \ldots, N_m$, where β_m is the protein-specific effect and ϵ is the error term. In the fitting procedure, we begin with the individual s_i's, and add interaction terms in a stepwise fashion using BIC as guidance.

2.3.2 Results and discussion

The GAM-based predictor is compared with two linear-based models. The first is the default weight coefficients in the Rosetta energy function; the second is a set of re-trained weights with a new linear regression run on the training set. Based on the three models, each decoy in the test set is scored, and the best scoring structure is chosen as the prediction. A GAM without interactions is also included for comparsion. Of primary interest is the frequency at which the native structure is ranked first; also of interest is the overall (average) ranking of the native structures. The results are

shown in Table 2.1.

Table 2.1: Ranking results on CASP9 test set.
Rosetta: Default weights in Rosetta energy function;
Regression: Refitted weights based on training set;
GAM+int.: GAM with stepwise-selected interactions;
GAM no int.: GAM with no interactions.

Protein ID	Rosetta	Regression	GAM+int	GAM no int.
515	6	5	1	4
516	3	10	2	5
517	4	5	1	3
518	3	3	1	3
520	9	4	1	6
521	6	23	3	23
522	6	18	9	2
523	6	5	1	1
524	3	1	1	2
525	4	2	1	1
526	3	1	2	2
527	9	4	1	4
528	3	2	1	4
529	2	2	3	3
530	5	8	2	2
531	12	3	1	1
532	3	6	3	5
533	5	9	1	6
534	2	3	1	3
536	6	9	1	3
537	1	2	3	3
538	7	14	3	3
539	8	4	1	2
540	6	1	1	1
541	7	27	1	2
542	4	2	1	3
543	3	2	3	3
544	7	7	1	1
545	7	6	1	2
547	5	2	2	4
548	10	3	1	2
549	10	46	37	45
550	3	2	1	3
551	12	4	1	1
552	5	4	1	2
553	8	4	1	2

Table 2.1: (continued)

Protein ID	Rosetta	Regression	GAM+int	GAM no int.
555	8	4	1	2
557	6	25	1	2
558	2	1	1	2
559	6	12	1	4
560	8	39	24	21
561	10	3	1	2
562	9	7	1	1
563	5	2	1	4
564	7	6	1	3
565	5	4	1	5
566	5	2	1	4
567	36	69	25	68
568	5	1	1	1
569	6	5	1	1
570	6	22	5	13
571	4	2	2	3
572	6	28	12	19
573	5	6	5	5
574	6	1	1	1
575	4	14	46	15
576	8	3	1	3
578	7	3	1	2
579	7	5	1	1
580	5	3	2	1
581	6	1	1	1
582	3	1	1	1
584	6	16	1	15
585	6	4	1	5
586	6	8	9	4
588	5	5	2	5
589	4	8	2	4
590	7	52	3	13
591	4	4	2	4
592	7	9	1	5
593	5	2	1	2
594	6	5	1	3
596	7	10	1	9
597	5	23	1	14
598	7	1	1	2
599	5	4	4	5
600	7	58	8	58
601	5	5	1	4

Table 2.1: (continued)

Protein ID	Rosetta	Regression	GAM+int	GAM no int.
602	17	25	6	28
603	5	7	1	4
604	4	2	1	2
605	14	27	33	24
606	5	7	1	2
607	3	7	1	3
608	4	6	1	4
609	4	1	1	3
610	5	4	1	2
611	7	8	1	8
612	8	4	1	8
613	6	4	1	4
614	6	1	1	1
615	7	23	2	14
616	12	10	1	8
617	9	19	10	18
618	6	16	2	9
619	6	16	1	3
620	6	8	1	7
621	6	1	1	2
622	6	3	1	1
623	5	7	1	3
624	5	5	1	2
625	4	11	1	4
626	3	25	1	7
627	5	3	3	7
628	4	1	1	2
629	5	18	33	14
630	6	4	1	1
632	10	11	1	7
634	6	19	1	10
635	5	32	1	10
636	4	13	4	5
637	6	31	1	28
638	6	22	23	10
639	8	12	1	1
640	5	31	2	34
641	4	3	1	2
643	6	19	7	21
Times native ranked first	1	13	79	18
Average rank of native	6.2	10.3	3.6	7.0

A few characteristics are worth noting. The Rosetta weights are built on different training data, which seem to be conservative in the sense of being optimal for the average ranking of natives: the native structure is rarely ranked first, but on average it is one of the best-scoring structures. A refitted linear regression on the training data is comparably worse, even though the native is identified more frequently; this illustrates sensitivity to the training set. The GAM, even without interactions, performs better than linear regression in both respects. It is comparable to default Rosetta weights for average ranking, but with higher variance. On the other hand, the GAM with selected interactions provides a substantial increase in the performance in both metrics, compared to the other three predictors.

This illustrates that there is indeed substantial overlap between energy components, which can be accounted for by a nonparametric smoother on interaction terms. This approach seems promising for the purpose of structure selection, after an appropriate sampling of the conformational space has been undertaken.

In passing, we note that the comparison described here computes energies for each decoy independently, and the selection is based a simple ranking of the decoy structure scores. It has been noted in the literature that leveraging similarities between competing structures (known as ensemble scoring) can further improve the identification of the best model among alternatives (Benkert et al., 2008). However, such an approach is unlikely to be helpful when there is a high degree of dependence in the decoy set. This dependence would be present in applications where the predictions are generated by a single algorithm, such as the type of Monte Carlo simulation to be considered in our work. This contrasts with overall ensembles in CASP experiments

where groups independently generate predictions.

2.4 Design of local moves

The top-level goal is Monte Carlo optimization of the energy of a template-based structure. A key ingredient is the generation of useful proposals during the simulation. These will come in the form of small perturbations to the structure at each step, followed by an evaluation of the energy function. Terms in any useful energy function will include summation over all pairwise atom distances (e.g. VDW) in the protein; thus, the computational bottleneck will be the energy evaluation if the entire-protein energy of structure proposals are required at each step. Therefore, effective proposals that only move small portions of the protein at a time are preferred; such proposals would require a partial energy calculation, namely that of the interactions between the modified region and the rest of the protein.

To this end, we have developed FRESS (Fragment Re-growth by Energy-guided Sequential Sampling), a sampling method for fragments within a protein with the two ends fixed. The method provides alternative conformations for backbone segments within the protein. A detailed description of FRESS is the subject of the next chapter.

A fragment sampler is a useful tool for refining loop regions of a protein, e.g. Jacobson et al. (2004), and we use FRESS as the main proposal mechanism in Monte Carlo simulations. When starting from a template-based model, sampling effort should be focused on regions with low template confidence. At each simulation step, a fragment proposal is evaluated with the energy function to determine acceptance or rejection. It is possible at this stage that closed fragment proposals

have minor steric clashes with the rest of the protein, but are otherwise good samples. Simply discarding the proposal would be an inefficient use of computational resources. We perform a quick check if steric problems in fragment proposals can be resolved, by adapting a version of the torsional relaxation technique, proposed by Wong et al. (1998).

In torsional relaxation, the goal is make small rotations to a torsion angle to reduce energy while minimizing the effect of the rotation to downstream residues. The rotation in one torsion angle is propagated by a series adjustments to neighboring torsion angles to gradually close the distance gap induced by the rotation. After the distance gap has been closed, the positions of residues further down the chain are no longer affected by the initial rotation. This method has been shown to be effective in stabilizing energy wells, without much atom movement in Cartesian coordinates. In the context of relaxing fragment proposals, we make the following adaptation.

Torsional relaxation for fragment proposals

Use FRESS to generate a closed fragment proposal, for residues $[l_0, l_0 + l]$

Relaxation loop:

For $i = l_0, l_0 + 1, \ldots, l_0 + l - 3$

Rotate ψ_i by 0.1° clockwise

Torsion relax this rotation

if torsion relax completes by residue $r < l_0 + l$

compute backbone energy change ΔE_{trial} on fragment $[i, r]$

if $\Delta E_{trial} < 0$ accept this relaxation

if torsion relax failed to complete or $\Delta E_{trial} > 0$

Rotate ψ_i by 0.1° counterclockwise

Torsion relax this rotation

if torsion relax completes by residue $r < l_0 + l$

compute ΔE_{trial} on fragment $[i, r]$

if $\Delta E_{trial} < 0$ accept this relaxation

if at least one relaxation accepted, goto Relaxation loop

This procedure is fast to execute relative to the FRESS sampling step, and helps to relieve steric strains without much position movement while making the fragment proposal to be favorable energetically.

Finally, a simple rotation of torsional angles is also considered as a local proposal, when the region is part an extended chain. Membership in an extended chain is determined by counting the number of residues that are within an 8 A radius.

The possible proposals for backbone atoms have now been described, and we turn our attention to side chains.

2.5 Optimization for side chains

An important consideration in all-atom protein folding is the prediction of side chains (Krivov et al., 2009). Thus, while FRESS and relaxation produces backbone

atom proposals, the side chains of the corresponding residues will also require optimization. The goal is to find an optimal side chain placement with the backbone fixed.

Side-chain packing has been a studied subject, e.g. Liang et al. (2011); Krivov et al. (2009); Canutescu et al. (2003); Xiang and Honig (2001); Tuffery et al. (1991). For this procedure, a side-chain energy function is required, to evaluate the impact of side-chain to side-chain and side-chain to backbone interactions. Liang et al. (2011) use the following functional form, derived from series expansions, to represent the energy of the j-th side chain dihedral angle of amino acid type i, when the angle in the configuration is α:

$$E(\alpha) \quad = \quad t_1^{i,j}\cos\alpha + t_2^{i,j}\sin\alpha + t_3^{i,j}\cos 2\alpha + t_4^{i,j}\sin 2\alpha + t_5^{i,j}\cos 3\alpha + t_6^{i,j}\sin 3\alpha,$$

where the set of parameters $\{t\}$ are optimized to distinguish native side chain conformations from decoys, using an objective function of the form (2.2). Additionally, the following form is used to represent the energy of the general atom-atom interaction between an atom of type i and an item of type j when they are separated by distance d:

$$E(d) = a_1^{i,j}d^{-2} + a_2^{i,j}d^{-4} + a_3^{i,j}d^{-6} + a_4^{i,j}d^{-8},$$

for the set of parameters $\{a\}$. These functional forms were shown to be flexible enough to capture angle and atom distance preferences, and provided good performance for side-chain prediction when a native backbone was given. A prediction is considered correct when the side chain dihedral angles are within 40° of the native value.

One appealing aspect of this approach is its amenability to optimization for continuous functions, and its use is not restricted to precomputed rotamer positions (Dunbrack Jr and Karplus, 1993). For use in FRESS fragment proposals, we perform a side chain optimization step after torsional relaxation. In addition to the current side chain angles, three starting side chain positions are sampled for each residue in the fragment from empirical side chain dihedral densities. Each of these sets forms an initial value for side chain energy minimization in Cartesian coordinates. Since the joint side-chain angle space for longer fragments will be of high dimension, the minimization is performed sequentially, by optimizing one the side chain of one residue at a time using the Levenberg-Marquardt algorithm (Moré, 1978). This is looped over the fragment until convergence, for each starting set of values. We then select the lowest energy side chains found. The goal is not to locate the global side-chain energy minimum for the fragment at this stage; this would not be computationally feasible if repeated over every backbone fragment proposal. However, our heuristic will generate side chains of sufficient quality to stabilize the energy of the fragment proposal.

2.6 Energy functions for sampling

We return to a discussion of energy functions, now in the context of sampling. With the use of the composite FRESS, relaxation, and side chain optimization method to generate complete fragment samples as local moves for a Monte Carlo simulation, it remains to specify the Boltzmann density under which the proposals should be evaluated.

The GAM approach that significantly improved structure ranking and selection might seem to also provide a basis for a pseudo-energy function for evaluating Monte Carlo proposals. However, we found that energy functions for sampling and selection must not necessarily be identical. Usage of the GAM-score as the pseudo-energy function for sampling failed to produce acceptable results, with the sampler being led astray by extraneous structures that were not physically sensible. This continued to be a problem after sampled decoys were added to training set and GAMs re-fitted in an iterative fashion. This provides empirical evidence that the conformational space is indeed too large for sampling on an artificial overfitted density to succeed; in every instance we were able to find structures that scored well on the complex model that were in fact poor conformations. In theory, this overfitting problem might be eliminated if representatives from the entire space were scored; this is not tractable in practice.

We therefore opt to use a standard linear combination of energy terms for a sampling energy function. The novelty in our heuristic approach is to account for the uncertainty in the coefficients of the terms, during sampling. Our objective function when training over a decoy set is to optimize the set of weights w_i to maximize

$$\sum_{j=1}^{M} Cor\left(\sum_{i=1}^{k} w_i E_{ij}, GDT_j\right),$$

that is, maximizing the average correlation of the linear combination with GDT, over the M unique training proteins. The maximal correlation is to encourage a funnel-type shape, such that structure scores improve as we sample closer to the truth. Weights were constrained to be positive, to preserve the ordering of energy values at

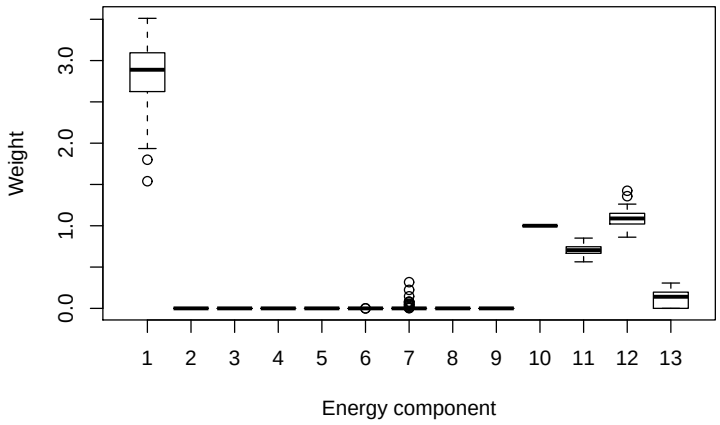


Figure 2.1: Cross validation on linear weights on 13 individual terms, for the design of sampling energy function.

the level of individual terms. The sidechain-sidechain and sidechain-backbone components of the DFIRE2 (Yang and Zhou, 2008) pairwise atom term (component 10) was set to be have a baseline weight of 1, to allow identifiability in the rest of the weights; this term was important and strongly positive in all replications. To characterize the uncertainty in the weights, we used a cross-validation approach to create confidence bands, where for each replicate 20% of the proteins were randomly held back as a test set. See Figure 2.1 for an illustration. Five of the 13 candidate components were deemed to be significantly non-zero based on this procedurre. To create sets of weights to be used in sampling, estimates from six folds of cross-validation were saved. These weight sets, along with a description of their corresponding terms, are shown in Table 2.2.

Since any sampling density is artificially imposed on the conformational space, one heuristic is to draw samples from an ensemble of densities representing the approxi-

Table 2.2: Weight sets estimated from cross validation. DFIRE2 sidechain and sidechain-backbone term has reference weight 1.

Description	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Van der Waals	3.00	2.66	3.21	3.15	3.43	2.58
Hydrogen bonding	0.14	0.02	0.14	0.17	0.22	0.15
Side chain torsion angle	1.04	0.99	1.01	1.03	1.06	1.10
DFIRE2 backbone-backbone	0.71	0.71	0.66	0.74	0.78	0.66

mate energy surface. Some of the local energy wells may overlap among the different weighting schemes; taken together, we expect a broader coverage of energy wells that may contain conformations of interest. To achieve this coverage, during sampling we allocate a fixed number of iterations to spend at each set of weights, and cycle through the different sets in this manner throughout the Monte Carlo simulation.

This approach inherently assumes that the any particular energy function is not completely reliable for sampling or selection. There is no longer a clear "minimum energy" conformation when samples have been drawn from an ensemble of energy functions. Therefore, our strategy is to save samples periodically, and use a different metric at the end of the simulation to select the best structure from those sampled. Based on the empirical success of GAMs for rank structures, this is the approach we adopt for the final selection.

2.7 Parallel samplers

It has been mentioned before that the conformational space of a protein is extremely large, even when constrained to refinement applications on template-based starting structures. Computation is the bottleneck, and it has long been recognized that some form of parallelization should be employed to locate energy optima faster

(Zhang et al., 2002) when the computing resources are available. Assuming that each CPU in the process is responsible for one Monte Carlo simulation thread, the parallelization scheme not only should search for candidate conformations individually, but the threads also should communicate to speed up global convergence to energy minima.

Parallel tempering is a natural candidate for this purpose, which Zhang et al. (2002) use as a starting point while recognizing its limitations. Therein, the authors suggest a more effective energy landscape flattening scheme than vanilla parallel tempering, by applying a nonlinear transformation $\tilde{E} = arcsh(E - E_0)$ for proposals with raw energy $E \geq E_0$, where E_0 is the raw energy of the current state and $arcsh$ is the inverse hyperbolic sine function. The intuition of this approach is to further flatten out the high-energy barriers (i.e. when $E >> E_0$) to improve sampling on the level of individual chains.

We believe the ideas explored in the development of the equi-energy sampler (Kou et al., 2006), can potentially provide a more efficient swaps than in parallel tempering. Draws from a neighboring chain will come from a similar energy level, and possibly also a different region of the space. Further work by Baragatti et al. (2012) extends equi-energy moves to the context of parallel tempering, called PTEEM (Parallel Tempering with Equi-Energy Moves). In our application, we opt for this approach since it inherits the advantage of equi-energy jumps, and make modifications so as not to require the pre-specification of energy rings and cutoffs. In particular, the expected range of possible energies and minimum energy for a given protein would be very difficult to estimate in advance; yet, that would be a required value for

implementing the full equi-energy sampler.

We briefly review the PTEEM algorithm, and discuss adaptations made in the context of our parallelization scheme for Monte Carlo optimization. As in the EE sampler, a sequence of $d+1$ energy levels are chosen, $H_1 < H_2 < \cdots < H_{d+1} = \infty$ with $H_1 = \min(E_{tot}(C))$ set to be the global minimum of the energy function over all conformations C. For N chains being run simultaneously, a sequence of N temperatures $T_1 = 1 < T_2 < \cdots < T_N$ are chosen; the i-th chain is draw samples from the Boltzmann density $\pi_i(C) \propto \exp(-E_{tot}(C)/T_i)$, $i = 1, \ldots, N$. The conformation space $\mathscr{C}$ is partitioned into energy rings

$$\begin{aligned} D_j &= \{C \in \mathscr{C}; E_{tot}(C) \in [H_j, H_{j+1})\}, \qquad j = 2, \ldots, d \\ D_1 &= \{C \in \mathscr{C}; E_{tot}(C) \in (-\infty, H_2)\}. \end{aligned}$$

These are the same rings for all chains, and these rings contain only current states. The global move is defined as follows: an energy ring D_j containing at least two chains is chosen; two chains within the ring are randomly selected and a swap between them is proposed.

The choice of energy ladder is not straightforward for proteins, as the energy range and minimum energy are unknown. Only the energy of the input template structure would be known. Thus the calibration technique suggested in the PTEEM paper, which involves running a vanilla MC chain to discover energy levels, would not be feasible. We instead opt to dynamically set and fine-tune the energy ladder.

First, we suggest that the number of rings d should be related to the number of chains N. If $d = 1$, the swaps are reduced to usual parallel tempering moves. If d is

only slightly smaller than N, when swaps are proposed there may not be many rings that contain current states for at least two chains. The heuristic we propose is setting $d \approx N/4$, so that on average rings will contain 4 states.

Second, the energy ladder will be fine-tuned as sampling proceeds and we gather more information on the energy landscape. For ease of implementation, global swap moves will occur at fixed intervals, with all chains swapping simultaneously when a global move is requested. Each time this step occurs, the energy rings are updated by setting a new H_1 and H_d, and then geometrically spacing the other energy levels between them. We use the following heuristics. Since $\min(E_{tot}(C))$ is unknown, we set H_1 to be the minimum energy of all sampled structures so far, plus some slack. A reasonable slack amount is the change in minimum energy found since the last global swap, if it is negative; this accounts for the overall trajectory of minimum energy over the simulation. The top energy ring, H_d, can be set at a level that accounts for both the maximum of the current states, as well as the maximum over all samples. If H_d is set to the maximum of all samples, it might include too much slack on the upper end as the simulation generally proceeds downwards in energy, leaving some top rings empty. As a heuristic we opt to set H_d to be a weighted average of the two maxima, according to their precision. Details are as follows.

Dynamic adjustment of energy rings for PTEEM

initialize empty vector E_{all}

do if PTEEM requests a global swap move:

let $E_{cur} \equiv$ vector of N current energies

append E_{cur} to E_{all}

$\Delta E = \min(\min E_{cur} - \min E_{all}, 0)$

$w_{cur} = \dfrac{1}{Var(E_{cur})}; \qquad w_{all} = \dfrac{1}{Var(E_{all})}$

$H_1 = \min E_{all} + \Delta E$

$H_d = \dfrac{w_{cur} \max E_{cur} + w_{all} \max E_{all}}{w_{cur} + w_{all}}$

geometrically space $H_2, \ldots, H_{d-1}$ between H_1 and H_d

2.8 Refinement in action

Thus far in this chapter, we have described the main tools and statistical considerations for the development a protein structure refinement algorithm. Here, we summarize the tools and put the pieces together.

Refinement begins by obtaining as input, usually through sequence alignment tools, an initial three-dimensional structure for the protein of interest. To generate an improvement on this starting structure, we take the approach of designing an energy function and Monte Carlo sampling algorithm to search the conformational space. The energy function used for sampling is a weighted linear combination of component energy terms, with different sets weights to account for the approximate nature of any energy calculation and enhance exploration. With the guidance of this energy function, Monte Carlo simulations are run with local proposal moves. These moves

involve a combination of FRESS fragment sampling, torsional relaxation, and rotation of torsion angles. Each proposal move will relocate portions of the protein backbone, usually resulting in misplaced side chains for the moved residues. We perform a local side-chain energy minimization on the affected residues, before sending the proposal to acceptance or rejection via the Metropolis ratio. The regions on which sampling effort are focuses will be guided by confidence in the structure template, if this information is available. Typically, a large set of simulation threads will be run simultaneously; to leverage the parallelization we adopt a version of parallel tempering with equi-energy moves to improve convergence. Samples are saved frequently, to build a set of possible structure predictions. Finally, at the end of the simulation, a GAM is used to rank the set of sampled structures, and the selected prediction is the highest ranking GAM score among the set.

In Figure 2.2, we see illustrated two examples of successful refinement, by applying this methodology.

2.9 Conclusions

Protein folding via computation continues to be a challenging problem after decades of research. In this chapter, we have only scratched the surface of the areas where more research is needed. We have identified a number of ways in which statistically-motivated computation can make a useful contribution, and proposed methods to tackle the protein structure refinement problem.

Many of the our suggestions involve the application of statistical intuition to learn from data more effectively and to sample more efficiently. One of the caveats when

dealing with large amounts of protein data and even larger conformational spaces, is that most proposed methodology will necessarily be heuristical in nature and difficult to test.

Our preliminary results show that there are areas of promise in these statistics-based approaches, though much future work awaits.

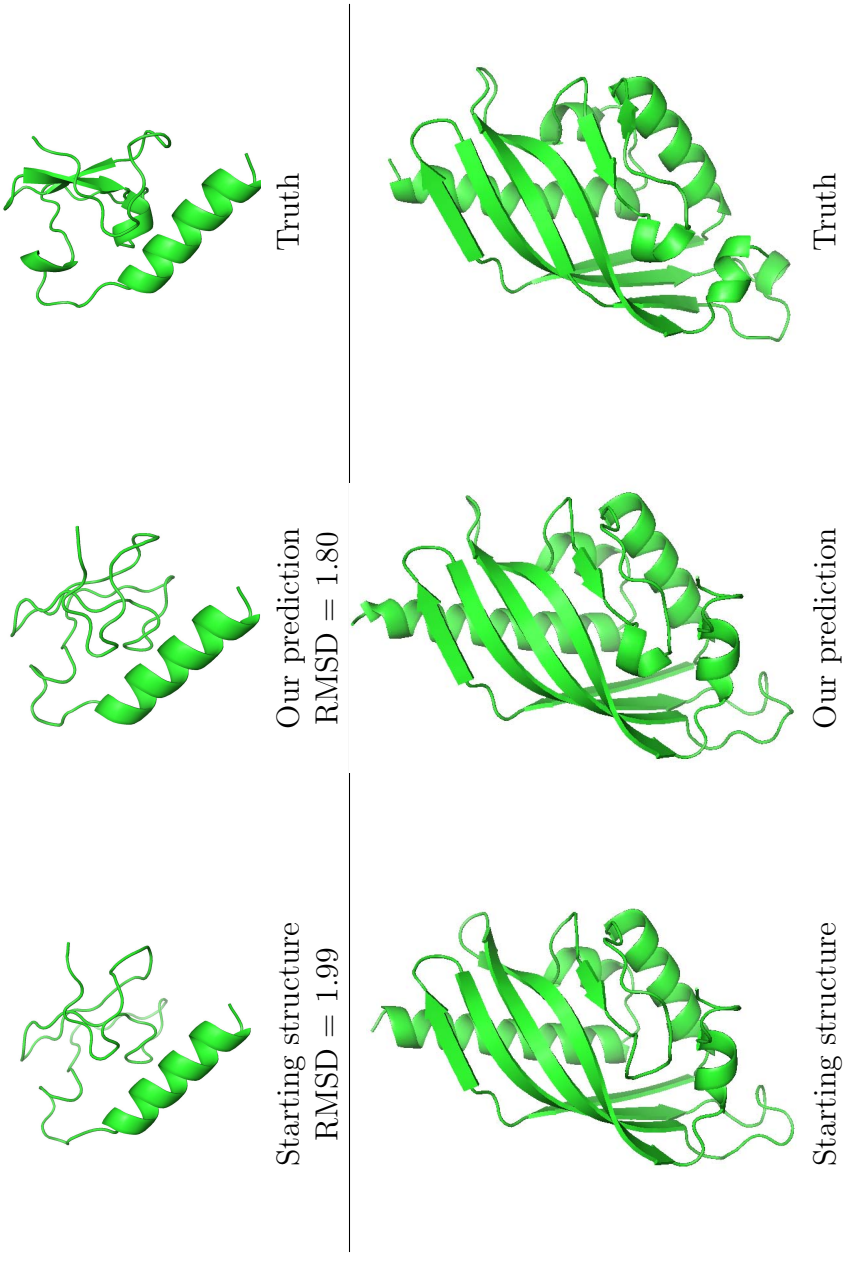


Figure 2.2: Two refinement examples.

Chapter 3

FRESS: A New Algorithm for Sampling Protein Fragments

An effective fragment closure algorithm is essential for loop modeling and also very useful in structure prediction and refinement. The statistical challenges of this problem are threefold. First, fragments of interest can be up to 18 residues or longer, which necessitates sampling from a high-dimensional space corresponding to the geometric degrees of freedom in the fragment. Secondly, there is a geometric constraint on fragment samples – they must begin and end at the designated positions (closure). Thirdly, the sampled fragments need to form favorable interactions with the remainder of the protein (feasibility) and have low energy. In this chapter we introduce a fragment sampler based on placing residues sequentially using sampling distributions designed to encourage closure and feasibility. We compared our method to existing ones such as Cyclic Coordinate Descent, CJSD, SOS, and FALC using a benchmark dataset for loop modeling. Based on these empirical assessments, we find

that our sampler is able to generate conformations with both low RMSD from the native conformation and good steric properties in a computationally efficient manner.

3.1 Introduction

Sampling the conformations of a fragment with its two ends fixed (fragment closure or fixed-end move) is an important problem in protein structure simulation and prediction. In loop modeling, conformations of fragments need to be sampled with other parts of the protein fixed (Fiser et al., 2000). In structure prediction, it can also be a very effective strategy for sampling protein conformations (Rohl et al., 2004b; Qian et al., 2007).

Fragment closure seeks to generate plausible spatial positions for the atoms of the fragment subject to fixed starting and ending points. Sampled fragments should satisfy two basic requirements. First, its bond lengths and angles must be compatible with the geometric constraints at its two ends. Second, steric clashes within the fragment and with other parts of the protein must be avoided, and favorable atomic interactions with low potential energy should be encouraged.

Designing fixed-end moves for chain polymers has been studied extensively for four decades (e.g., Go and Scheraga, 1970; de Bakker et al., 2003; Betancourt, 2005; Cahill et al., 2003; Canutescu and Dunbrack, 2003; Collura et al., 1993; Coutsias et al., 2004; Cui et al., 2008; Jacobson et al., 2004; Koehl and Delarue, 1995; Lee et al., 2005; Liu et al., 2000; Lee et al., 2010; Liu et al., 2009; Mandell et al., 2009; Moennigmann and Floudas, 2005; Noonan et al., 2005; Peng and Yang, 2007; Sellers et al., 2008; Shenkin et al., 1987; da Silva et al., 2004; Soto et al., 2008; Uhlherr, 2000; Uhlherr et al., 2001;

Vendruscolo, 1997; Wedemeyer and Scheraga, 1999; Wick and Siepmann, 2000; Xiang et al., 2002; Zhang et al., 2007a; Zhu et al., 2006). Most existing fragment closure methods can be divided into three categories. The first type of methods, including PLOP, make small changes for several adjacent torsion angles (Cahill et al., 2003; Cui et al., 2008; Go and Scheraga, 1970; Noonan et al., 2005; da Silva et al., 2004; Wedemeyer and Scheraga, 1999; Zhu et al., 2006). Due to the local nature of this type of move, it often takes many steps to change a conformation to a significantly different one, requiring a high computational cost to sample the large and contrained conformational space of proteins. The second type of methods, such as CCD, SOS, FALC, Loopy, and CJSD, employ a two-step approach to closure (Canutescu and Dunbrack, 2003; Collura et al., 1993; Coutsias et al., 2004; Koehl and Delarue, 1995; Lee et al., 2005, 2010; Liu et al., 2000, 2009; Mandell et al., 2009; Shenkin et al., 1987; Soto et al., 2008; Xiang et al., 2002). First, they place backbone atoms without respect to geometric constraints. Next, a deterministic or random procedure morphs this initial conformation into one that satisfies the geometric constraints. The ability to make larger conformational changes in a single step helps these methods achieve higher efficiency. The third type of methods close fragment conformations using a chain growth approach by placing residues/atoms one at a time with some constraints to favor the closure (de Bakker et al., 2003; Jacobson et al., 2004; Vendruscolo, 1997; Wick and Siepmann, 2000; Zhang et al., 2007a). Chain growth methods can make large conformational changes in a single step and can potentially be quite efficient. The problem faced by chain growth methods is how well they can "foresee the likelihood" of an early placement for eventually producing connected conformations. The

proposed algorithm in this paper belongs to the chain growth group.

Fragment closure is often the starting point of loop modeling, where the end goal is an *ab initio* construction of a fragment prediction that resembles the native conformation as closely as possible. As the length of the loop increases, the space of possible loop conformations becomes very large (Jacobson et al., 2004; Zhu et al., 2006; Zhao et al., 2011). Thus, it is important for a fragment closure algorithm to produce a set of conformations with high probability of containing at least one that is similar to the native conformation. The early identification of promising loop conformations becomes crucial, for further loop refinement and selection to succeed using these as starting conformations (e.g., Zhao et al., 2011). This goal should be achieved in a computationally efficient manner.

The complete energy function used to score and select loops often cannot be applied to all loops found during the sampling phase, as energy evaluation will dramatically increase the computational time required. However, if steric feasibility is entirely ignored during sampling, the quality of sampled conformations will often be quite low with many atomic clashes, as fragments are normally surrounded by atoms from other parts of the protein. Several recent studies (Zhang et al., 2007a; Soto et al., 2008; Liu et al., 2009) have taken steric and other atomic interactions into account during the conformation sampling step and were able to generate fragment conformations of higher quality with improved overall performance. In this study, we investigate whether the performance of this type of method can be further improved with more sophisticated sampling strategies, especially at longer loop lengths.

The method presented in this paper is called Fragment Re-growth by Energy-

guided Sequential Sampling (FRESS), based on the efficient fragment closure algorithm that was first developed and tested on hydrophobic-polar (HP) models (Zhang et al., 2007a). The FRESS method achieved significantly better performance than previous methods on benchmark HP sequences. This paper introduces an implementation of the FRESS method on off-lattice protein models with a Cartesian coordinate representation of protein structures. To achieve high closing efficiency for real proteins, we develop more advanced proposal distributions that take several constraints into account during the sampling step. There are no required energy-minimizing steps after the sequential residue placement. In practice, we can obtain improved steric properties of our samples by applying a fast torsional relaxation step (Wong et al., 1998), which minimizes the energy locally for a fragment that is already closed. This yields significant performance gains compared to the two-step type of methods mentioned above, which achieve steric feasibility through post-hoc minimization. The actual closure rate obtained is an order of magnitude higher than using only torsion angle sampling (de Bakker et al., 2003).

Another advantage of our method is that it can be used for generating proposals in a larger Markov Chain Monte Carlo (MCMC) simulation for structure prediction. FRESS provides a way to evaluate the Rosenbluth weight (Rosenbluth and Rosenbluth, 1955) of a regrown fragment (i.e., its probability of acceptance). Most other fragment sampling methods are incompatible in this regard because they cannot evaluate the proposal density of their samples.

We test our sampling method on benchmark loop modeling datasets. For loops of length four to 12, the criteria we examine is the minimum RMSD in 5000 sampled

fragment conformations within a given computational time budget, to allow comparison with other methods. For the set of longer loops of length 14 to 17 residues, we test our method's ability to produce a promising set of of low RMSD initial loop conformations for further modeling.

In the rest of the paper, we develop the theoretical framework of FRESS and the residue sampling distributions in Section 3.2. We test the method on benchmark loop modeling datasets and compare with other previous methods in Section 3.3. We conclude the paper with a brief discussion in Section 3.4.

3.2 Methods

Our method attempts to place each residue in a fragment sequentially, in contrast to the "fragment assembly" approach of sampling entire fragments from known conformations (e.g. Lee et al., 2010; Rohl et al., 2004b). The challenge with sequential placement, especially for longer fragments, lies in attaining closure and feasibility, while at the same time efficiently exploring the space of low energy conformations. FRESS is inspired by the configurational bias Monte Carlo (CBMC) method (Siepmann and Frenkel, 1992; Frenkel et al., 1992) to encourage the placement of each residue to achieve steric feasibility and increase the probability of successful closure. The key idea is to allow the sampler to learn from the environment as sequential placement occurs.

Throughout, we adopt a number of conventions standard to loop modeling. We limit our focus to the construction of the backbone, since side chains are usually placed during loop scoring and selection (Bonneau and Baker, 2001). Bond lengths

and angles are assumed fixed to the standard values of Engh and Huber (1991), with the exception of the C_α bond angle. Finally, the fragment is assumed to start with the backbone carbonyl C atom of the initial residue and end with the C_α atom of the final residue. The terminal C atom refers to the backbone carbonyl C atom in the final residue, which is fixed in space and considered the target for closure. The global RMSD is calculated when comparing sampled loops to the native conformation.

3.2.1 Formalization

This section formalizes the FRESS procedure. Let l be the number of residues in the fragment to be sampled. To obtain each sampled fragment, FRESS sequentially grows the residues $R_i,\ i = 1, \ldots, l-2$, and then attempts to close the fragment using the fast analytical closure method of Coutsias et al. (2004). This is because the placement of the final two residues is essentially deterministic when closure is required, due to four geometric constraints at the closing seam of the fragment: the closing bond length; the two closing bond angles; and the ω torsion for the first C_α atom outside the fragment (Go and Scheraga, 1970). Closure is not always possible, e.g. if the distance from the $(l-2)$-th grown C_α to the closing C_α is too small or too large, and/or the growth up to the $(l-2)$-th residue is misoriented in direction. FRESS thus seeks to grow residues in a manner that increases the likelihood of successfully obtaining closure after the $(l-2)$-th residue has been reached.

The growth of each residue requires the sampling of its relevant geometric degrees of freedom. The backbone torsion angles ϕ and ψ per residue account for most of the diversity possible in backbone growth. The torsion angle ω, while usually close to π,

also has a small degree of flexibility in native conformations. Of the bond angles, the N–Cα–C angle τ, while usually close to 110° can sometimes stretch to relieve steric strains (Karplus, 1996). We were able to accurately rebuild a large collection of native conformation backbones by allowing these four geometric aspects to vary, while fixing the other bond lengths and bond angles to the standard values reported in Engh and Huber (1991). We found that 5° increments in ϕ and ψ provided sufficient resolution for reproducing native structures.

To build the i-th residue, FRESS will sample the set of values $R_i = (\phi_i, \psi_i, \omega_i, \tau_i)$. Suppose that there is a known energy function E, which accounts for interactions both within the fragment and between the fragment and the fixed part of the protein. The FRESS fragment proposal sampling procedure requires a series of conditional distributions; at each residue we require a conditional distribution of the i-th residue given the previous ones (which we denote $R_i|R_{<i}$ for short). Each conditional draw of a residue will have two components, which promote closure and steric feasibility respectively. Let $s_i(R_i|R_{<i})$ denote the distance-based component that encourages closure, and $E_i(R_i|R_{<i})$ denote the incremental energy that measures the steric impact of residue R_i given the previous ones.

The ϕ_i and ψ_i backbone torsion angles are the main drivers of diversity in backbone growth. As a result, the first step in obtaining R_i involves the sampling of ϕ_i and ψ_i from a torsion angle map with 5° by 5° grid resolution, and ω_i and τ_i fixed at ideal values. Suppose there are k bins with non-zero probability in the trial distribution, i.e. $s_i(R_i|R_{<i}) > 0$; these bins represent the directions in which residue growth may possibly proceed. Next, the incremental energy E_i is then computed for these bins

and combined with s_i to create a final sampling distribution for ϕ_i and ψ_i. From this distribution, we make d independent draws of (ϕ_i, ψ_i). The construction of s_i and the combined $s_i \times E_i$ distributions will be described in detail in Section 3.2.3. Finally, for each of the d draws of (ϕ_i, ψ_i), ω_i and τ_i are sampled from Gaussian distributions centered at ideal values with a small variance in accordance with that observed in native conformations. As a result of this procedure, d draws of R_i are obtained.

Since residues are grown one a time, the fragment being built may run into "dead ends" over its course of construction where one of the following two conditions occur when sampling a residue: (1) $k = 0$, that is, there are no bins with non-zero probability in the trial distribution, indicating that eventual closure of the fragment will not be possible; (2) the incremental energy is high for all k bins, indicating that a steric clash is unavoidable if growth is to continue. To increase the efficiency, FRESS includes a pruning step where partially grown fragment conformations will be terminated early if either of these conditions occur. In this case, having $d > 1$ draws saved per residue during placement allows FRESS to back up to an earlier portion of the fragment and resume growth.

More precisely, the FRESS procedure for sampling the i-th residue can be written as follows:

1. Compute the distribution $s_i(\phi_i, \psi_i, \omega_i = 180°, \tau_i = 111°|R_{<i})$,
   over the grid $\phi_i, \psi_i \in \{-175°, -170°, \ldots, 180°\}$.

2. Let $(\phi_i^{(1)}, \psi_i^{(1)}), \ldots, (\phi_i^{(k)}, \psi_i^{(k)})$ denote the k grid bins with $s_i(R_i|R_{<i}) > 0$.

3. Compute $E_i(\phi_i^{(j)}, \psi_i^{(j)}, \omega_i = 180°, \tau_i = 111°|R_{<i}),\ j = 1, \ldots, k$.

4. Sample d pairs of (ϕ_i, ψ_i) independently from the probability mass function

$$P[(\phi_i, \psi_i) = (\phi_i^{(j)}, \psi_i^{(j)})] = \frac{s_i(R_i^{(j)}|R_{<i}) \exp(-E_i(R_i^{(j)}|R_{<i})/T)}{\sum_{j=1}^{k} \left\{ s_i(R_i^{(j)}|R_{<i}) \exp(-E_i(R_i^{(j)}|R_{<i})/T) \right\}},$$

$j = 1, \ldots, k$, where $R_i^{(j)} = (\phi_i^{(j)}, \psi_i^{(j)}, \omega_i = 180°, \tau_i = 111°)$, and $T > 0$ is the temperature chosen for the Boltzmann distribution corresponding to E_i.

5. Draw d samples from

$$\omega_i \overset{\text{iid}}{\sim} N(179.3°, 4.1°)$$
$$\tau_i \overset{\text{iid}}{\sim} N(110°, 3.5°)$$

As noted earlier, the function E_i should be chosen to encourage the sampling of sterically feasible conformations, without being too computationally intensive. A full energy function for fragment scoring and selection would be overly burdensome, and unlikely to be helpful at this stage before the full chain is grown and side chains are added. As a reasonable trade-off, our E is composed of two parts: (1) A local Van der Waals energy function based on the OPLS-AA force field parameters (Kaminski et al., 2001); (2) the hydrophobic term introduced in (Zhu et al., 2006) to encourage the burial of grown backbone C atoms. There is much flexibility in terms of which scoring function to employ for this step. To keep computation costs low, E_i should only involve interactions between the residue being placed and the rest of the protein.

An important free parameter in FRESS is d, the number of samples saved at each residue. With larger values of d, we are more likely to sample a feasible, closed

fragment by pruning and backing up from dead ends to resume growth. However, if d is too large, we might waste computational time exploring non-promising portions of the conformational space when the early residues are poorly placed, so there is a tradeoff. The nature of this relationship is explored later in Section 3.3.

3.2.2 Monte Carlo sampling

Consider a typical folding simulation for an entire protein, in which we seek to optimize a whole-protein energy E over the space of configurations represented by backbone torsion angles $A_i,\ i = 1,\ \ldots,\ n$. A Monte Carlo optimizer, such as simulated annealing, typically requires sampling from the Boltzmann distribution at a temperature $T > 0$,

$$p(A) \propto \exp(-E(A)/T),\ A \in (-\pi, \pi)^n,$$

Samples from p can be generated using the Metropolis-Hastings algorithm for any proposal distribution, so long as its proposal density can be evaluated to ensure detailed balance. With appropriate modifications, FRESS can act as a valid proposal generator to modify the chain. For instance, Zhang et al. (2007a) suggest randomly selecting and regrowing fragments to explore the energy landscape in *ab initio* folding.

We briefly describe the computation of Rosenbluth weights in the context of FRESS. These weights ensure detailed balance, as shown by (Frenkel et al., 1992) in the development of CBMC. Suppose a fragment with original residues $R_{0,1}, \ldots, R_{0,l}$ is sequentially regrown to obtain $R_{new,1}, \ldots, R_{new,l}$, with $j_i \in \{1,\ \ldots,\ k_i\}$ denoting the index of the chosen draw for the i-th residue, when k_i draws are made from

$s_i(R_i|R_{<i})$. The weight $Q(\alpha)$ is given by the product of the selection probabilities over the l residues in the sampled fragment. The weight $Q(\alpha_0)$ corresponds to the same product, but computed over the original fragment. Specifically,

$$\begin{aligned} Q(\alpha) &= \prod_{i=1}^{l} \frac{\exp(-E_i(R_{new,i}^{(j_i)})/T)}{\sum_{j=1}^{k_i} \exp(-E_i(R_{new,i}^{(j)})/T)} \\ Q(\alpha_0) &= \prod_{i=1}^{l} \frac{\exp(-E_i(R_{0,i})/T)}{\exp(-E_i(R_{0,i})/T) + \sum_{j \neq j_i} \exp(-E_i(R_{new,i}^{(j_i)})/T)} \end{aligned}$$

The regrown fragment is then accepted with probability

$$min\{1, Q(\alpha)/Q(\alpha_0)\}.$$

The computation of this extra weight has a non-negligible impact on the overall speed of a long simulation. Therefore, when Monte Carlo optimization of the energy function is desired, rather than samples from the stationary distribution, we recommend omitting the Rosenbluth weight for greater efficiency.

3.2.3 Construction of residue sampling distributions

The construction of the (ϕ, ψ) torsion angle distribution for s begins with the Ramachandran density plot (Ramachandran et al., 1963) for the given residue and secondary structure type, gridded into 5° by 5° bins. We denote this distribution as s_0. For most residue types, this results in roughly 800 to 1000 bins with non-zero probability. Sampling (ϕ, ψ) from this grid alone will generate realistic torsion angles,

but does not guarantee that the fragment will grow towards the closure target.

The main idea of our method is to update s as each residue is being placed, such that the incremental approach and remaining distance to the closure target are empirically sensible. To achieve this goal, a series of empirical distributions is constructed to help guide sequential residue placement. These distributions help encourage fragment closure while learning from the growth environment.

There are two aspects of distance information to incorporate into the s distribution. Firstly, the remaining distance to the terminal C_α anchor should be empirically reasonable after the residue is placed. This is done by an empirical tabulation of joint C–C_α and C_α–C_α distances, conditioned on three factors: the atom type (C or C_α), the residue separation (from two to 18), and the secondary structure of the residue (helix, sheet or coil). Secondly, the incremental distance towards the terminal C_α should be sensible, given the remaining distance to the terminal C_α from the previous residue. This helps to avoid directional misorientation during growth which can prevent the fragment from closing.

More formally, suppose that we are currently sampling the i-th residue in a fragment of length l. Let d_0 be the distance from the $(i-1)$-th C_α to the l-th (i.e. terminal) C_α. For a given (ϕ_i, ψ_i), with ω_i and τ_i fixed at their ideal values, the backbone coordinates of the $(i-1)$-th C atom, and the i-th N and C_α atoms are determined. Then let d_1 be the distance from the $(i-1)$-th C to the l-th C_α, which depends on the choice of ϕ_i (since τ_i is fixed). Let d_2 be the distance from the i-th C_α to the l-th C_α, which depends on the choice of ψ_i (since ω_i is fixed). Thus, the sampled pair of torsion angles (ϕ_i, ψ_i) maps to the distance pair (d_1, d_2). The

effect on distance by varying ω_i and τ_i at this step would be negligible. With these definitions, the first aspect of distance information to incorporate is to ensure that (d_1, d_2) is sensible; the second aspect is to ensure that $(\Delta d_1, \Delta d_2) \equiv (d_1 - d_0, d_2 - d_0)$ is sensible.

The values $(\Delta d_1, \Delta d_2)$ capture a simple notion of current growth direction in the fragment. Positive values of $(\Delta d_1, \Delta d_2)$ indicate that atoms of the sampled residue are growing away from the terminal C_α, while negative values of $(\Delta d_1, \Delta d_2)$ indicate growth toward the closure target. Intuitively, negative values of $(\Delta d_1, \Delta d_2)$ are favored at higher distances of d_0. This helps to move the chain towards the terminal C. The effect is the greatest when only a few residues remain to be placed. See Figure 1 for an illustration.

To build these empirical distance distributions, we compute statistics across the database of 16,482 proteins from the daily `bc-30` list on the Protein Data Bank on January 18, 2010. The list contains a single representative from each of 16,482 groups clustered using BLASTclust (Altschul et al., 1997) so that no proteins share greater than 30% sequence similarity across groups. Secondary structure information is either obtained directly from PDB files or estimated using DSSP (Dictionary of Secondary Structure of Proteins), as developed by Kabsch and Sander (1983). Missing atoms and residues are dropped from all empirical statistics.

Let $f_{l-i}(d_1, d_2)$ denote the joint density of C–C_α and C_α–C_α distances, for a residue separation of $l - i$. Let $g_{l-i}(\Delta d_1, \Delta d_2 | d_0)$ be the corresponding joint density of distances that captures the growth orientation. Based on statistics across the PDB, we approximate the distance distributions by binning the central 99.9% of the empirical

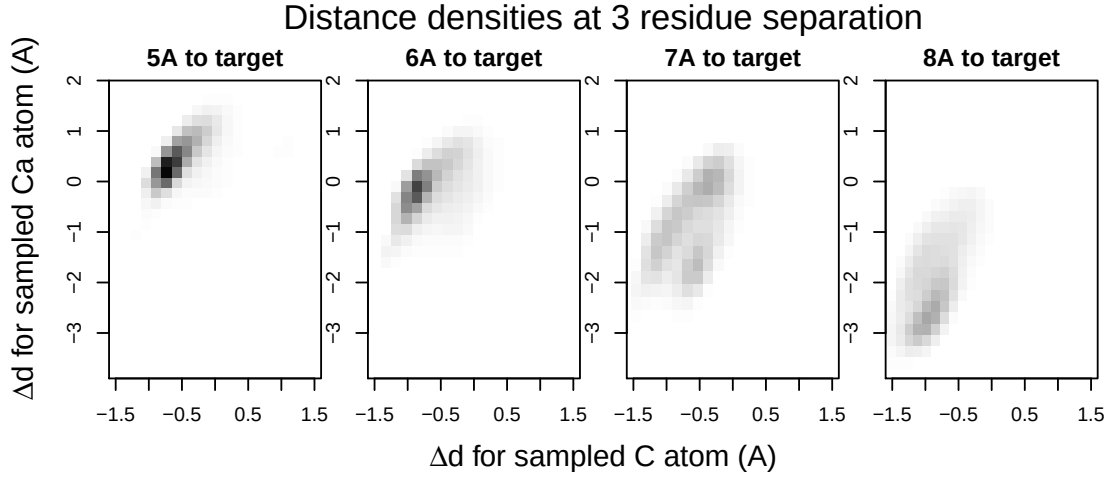


Figure 3.1: Joint approach distance distributions for sampling C, C_α atoms of the current residue, when there are 3 residues remaining in the fragment. The distributions are conditioned on the remaining distance to the target, from the C_α of the previous residue. Δd is defined to be the change in remaining distance to the target, for the corresponding C, C_α atoms of the residue when placed. The densities shift to reflect the range of feasible Δd from which to sample. For instance when the previous C_α is 8 A away, sampling for the current residue becomes strongly guided to close the distance gap, so that we are 0.2 to 3.6 A closer to the target after the C_α is placed.

distances into a 16-by-16 histogram grid, followed by smoothing across adjacent cells. For construction of the g distributions, the conditioned distance d_0 are rounded to the nearest Angstrom.

For sampling, the torsion angle distribution s must then be updated by the distance probabilities, to promote both realistic torsion angles as well as favorable distance properties. The distances will be converted to torsion angles so that the densities can be multiplied. Let s_0 be the raw empirical binned distribution of torsion angles (from the Ramachandran map) and f, g of distances as defined above. Let $D : (-\pi, \pi) \times (-\pi, \pi) \mapsto \mathbb{R}^+ \times \mathbb{R}^+$ map a torsion angle pair (ϕ, ψ) to the distances (d_1, d_2), and ΔD map the same angles to the incremental distances $(\Delta d_1, \Delta d_2)$. Then our final specification of s_i can be written as

$$s_i(\phi_i, \psi_i) \propto s_0(\phi_i, \psi_i) \times [f_{l-i}(D(\phi_i, \psi_i))|J| \; g_{l-i}(\Delta D(\phi_i, \psi_i)|d_0)|J|]^\lambda$$

where f and g are probability mass within the element of the 16-by-16 distance grid, to which the torsion angles map. J refers to the Jacobian of the distance-to-angle map. Finally, λ controls the relative weight of the distance component.

Recall that our procedure includes a pruning step. In the context of this section, if the empirical distance grid has no mass in the entire range of torsion angles, it means that no fragment has ever closed when the current residue is as far away from the terminal C as it is at the present stage, and the growth is terminated.

Figure 2 illustrates the process of updating the (ϕ, ψ) sampling distributions. The raw torsion angle map is shown in the leftmost panel. The middle panel shows the updated distribution after incorporating the distance component. The final samples

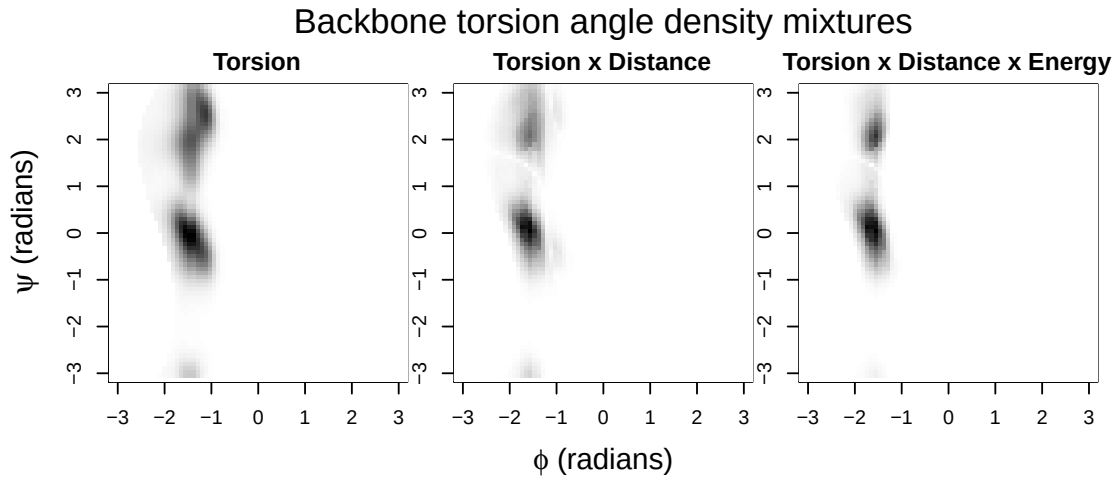


Figure 3.2: Construction of ϕ, ψ sampling densities. The leftmost panel shows the Ramachandran map for pure torsion angle sampling s_0; these are the allowable torsion angles for the residue. The center panels shows the updated density s_i in the ϕ, ψ space, after incorporating the distance distribution. The rightmost panel shows the final density after mixing in the incremental energy function ($s_i \times E_i$). Note how the relevant angle space becomes restricted as these elements are added. The final d samples of ϕ, ψ for this residue then drawn from this map.

for (ϕ, ψ) are drawn after further multiplying s_i with the energy component.

3.3 Results

For the purposes of loop modeling and structure prediction, the key assessment criteria for a fragment sampler are two-fold: (1) how close its samples are to the native conformation; (2) its computational efficiency in drawing good samples. While an algorithm's overall speed in generating closed loops can be a useful intermediate metric, the value of samples for loop prediction is finally limited by the closeness to the native conformation of the best sampled loops. For long fragments in particular,

the sampling of a large number of poor, sterically infeasible loops is undesirable due to the added computational burden of screening and refining the loops using a full energy function during the prediction phase.

We thus proceed with the evaluation of the proposed method on loop datasets using a combination of two metrics: (1) the minimum global root mean squared deviation (RMSD) between 5,000 samples and the native, calculated on the backbone atoms C, C_α, N, and O; (2) the best RMSD loop sampled within a given computational time budget. The first metric allows the comparison of FRESS performance with existing fragment samplers for loops of length four to 12. The second metric is important when FRESS is to be used as proposal generator for a larger MCMC sampling problem, for example when loops from an entire protein need to be optimized.

Two loop datasets are used for this section. The first is the set 30 loop targets of lengths 4, 8, and 12 in Canutescu and Dunbrack (2003). The second is the the set of 89 long loop targets of lengths 14 to 17 in Zhao et al. (2011). The first set allows comparison against other methods based on the 5,000 sample metric. The long loop dataset shows the potential of FRESS for generating promising samples for the initial phase of long loop modeling. Since most loop sampling methods are closed-source, we compare against the results shown in Soto et al. (2008) where possible. For the purpose of sampling these loops, FRESS uses distance and torsion angle distributions that were built for secondary structure of coil type.

The important settings for FRESS are the following: (1) the temperature T associated with the energy function; (2) the limit on the incremental energy E_i per residue for pruning; (3) the weight λ that controls the relative importance of the torsion and

distance distributions; (4) d, the number of draws saved at each residue.

To allow for computational flexibility and transparency, we attempt to make most of these settings adaptive. The temperature T controls the degree to which low energies are favored when growing each residue. The energy landscape and local steric environment for each loop is different, and the temperature should be set to allow a variety of loops to be generated. A sampler that produces a similar closure every time is not useful because it wastes computational time without increasing the probability of locating near-native conformations. We thus set FRESS to automatically alternate between a range of temperatures $1 \leq T \leq 16$ as it samples fragments.

The limit on per-residue incremental energy E_i reflects a trade-off between closure rate and the overall steric feasibility of sampled fragments. Sampled residues that exceed this limit are pruned. In native conformations, the VDW backbone energy on a per-residue basis is usually low (< 5 kcal/mol). Since residues are grown one at a time, imposing a similar constraint during sampling will generate fragments with near-native VDW energies, at the expense of a low overall closure rate since many draws will be pruned. To achieve a balance of closure and low energy, FRESS dynamically monitors the number of closed fragments per trial. The initial limit on incremental VDW energy is set to 5 kcal/mol. After a burn-in period, an adjustment is made whenever the closure rate drops below 10%. The maximum VDW increment is successively increased in 1 kcal/mol steps to target a closure rate of 10% or higher.

The setting λ controls for the weight of the distance component in s_i relative to the raw torsion angle distribution s_0. The distance distributions become increasingly important as growth nears the end of the fragment. When analytical closure is applied

for the final two residues, the end of the partially grown fragment should be situated at a distance and orientation that is favorable for closure to occur. To encourage this behavior, we apply a set of weights λ that increase linearly over the fragment. These range from a weight of $1/l$ relative to the torsion distribution at the beginning of the fragment, up to an equal weight with the torsion distribution for the final residue grown before analytic closure is applied.

Finally, the setting $d \geq 1$ is the number of draws saved at each residue. If $d = 1$, the current trial is terminated as soon as pruning occurs. If $d > 1$, then upon pruning FRESS backs up to a previous residue where the growth is still feasible, and uses one of the alternative draws at that residue to resume growth. The current trial is only terminated when all d draws are recursively exhausted. The effect of $d = 1, 3, 5$ is shown in Table 3.1, for sampling the length 12 loops in Canutescu and Dunbrack (2003) with a fixed CPU time budget of 15 minutes. Higher values of d lead to higher closure rates, at the cost of higher computational time per trial. Taking $d = 3$ is a reasonable point of compromise, generating the most closed loops per unit time.

To compare with other methods for loops of length eight and greater, FRESS is run with a given computational budget on a 2.1 GHz CPU. Often, fewer than 5,000 samples that satisfy the steric feasibility screens are produced when the time budget is short. To allow a strict comparison with other methods, the 50 FRESS samples with the best VDW energy after applying backbone torsional relaxation from (Wong et al., 1998) are chosen as seeds for enrichment to generate a total of 5,000 samples. The enrichment occurs by randomly perturbing the ϕ and ψ torsion angles of these samples by $\pm 5^\circ$ increments. Fragments that can analytically close and have low VDW

Table 3.1: Number of feasible samples drawn and proportion of fragments closed, for different values of d given 15 minutes of CPU time on loops of length 12.

	$d = 1$		$d = 3$		$d = 5$	
Loop	Samples	Closed	Samples	Closed	Samples	Closed
1cruA_358	247	0.06	193	0.20	39	0.22
1ctqA_26	334	0.08	709	0.72	590	0.94
1d4oA_88	297	0.05	719	0.63	627	0.88
1d8wA_46	192	0.06	398	0.43	258	0.31
1ds1A_282	240	0.05	136	0.33	12	0.30
1dysA_291	234	0.07	546	0.47	438	0.67
1eguA_508	219	0.05	273	0.43	175	0.73
1f74A_11	234	0.07	350	0.38	313	0.37
1qlwA_31	487	0.10	770	0.74	750	0.89
1qopA_178	633	0.18	626	0.75	243	0.81
Average	312	0.08	472	0.49	345	0.61

energy after perturbation are added to the list of samples.

Table 3.2 shows the results for the 30 loop targets of lengths 4, 8, and 12 in Canutescu and Dunbrack (2003), based on minimum RMSD attained in the first 5,000 closed feasible samples, with FRESS limited to 15 minutes of CPU time. FRESS is compared with the following methods: Ramachandran map CCD (Canutescu and Dunbrack, 2003), the CSJD method (Coutsias et al., 2004), the SOS algorithm (Liu et al., 2009), the FALC/FALCm methods (Lee et al., 2010); RMSD values for these are taken from Table II of Lee et al. (2010). FRESS has comparable resolution at the shortest fragments of length four and eight; for the longer length 12 loops, FRESS has the most favorable performance. The advantage of using our residue sampling distributions becomes more apparent as loop length increases.

Next, we show FRESS results for the set of 89 long loop targets of lengths 14 to 17 studied by Zhao et al. (2011) in Table 3.3. Results for other methods are not

available for sampling loops of these lengths. We note that FRESS is able to produce conformations within the resolution required for further loop refinement as in Zhao et al. (2011), with the RMSD metric continuing to perform well for these longer loops without significant degradation.

3.4 Conclusion and discussion

In summary, we have developed an efficient fragment closure method using sequential sampling. Residue sampling distributions are constructed to increase the efficiency in fragment closure and exploration of low energy conformational spaces. The method was tested on benchmark loop modeling datasets and performs better than earlier methods in the criteria we have examined. Our method is based on growing residues sequentially with favorable empirical distances from the fixed terminal anchor and favorable torsion angles, while accounting for steric feasibility via a simple energy function. It does not require a post-hoc closure step to achieve closure, and is able to generate good fragment samples efficiently.

FRESS is also highly extensible. The version discussed here conditions on the distance to the terminal C and secondary structure. It is possible to extend this by incorporating additional conditioning variables, such as residue or solvent accessibility. We have also implemented FRESS as a proposal step within a larger tertiary structure prediction system to refine homology models based on repetitively resampling fragments to find lower energies (Zhang et al., 2007a). Our sequential strategy takes a high-dimensional sampling problem and divides it into tractable pieces; in this case an appropriate unit is one residue at a time. The constraints imposed by

Table 3.2: Comparison of fragment sampling methods' minimum RMSD from native reached after 5,000 samples, for the set of 30 loops in Canu et al (2003). Results for the first five columns are taken from Table II of Lee (2010). For 8 and 12 residue loops, a computational time limit of 15 minutes is imposed on FRESS. For the purpose of this comparison, the sampled loops in FRESS are enriched to create a total of 5,000 closed loops (see text for explanation).

Length	Loop	CCD	CJSD	SOS	FALC	FALCm	FRESS
4-res	1dvjA_20	0.61	0.38	0.23	0.34	0.39	0.33
	1dysA_47	0.68	0.37	0.16	0.17	0.20	0.17
	1eguA_404	0.68	0.36	0.16	0.22	0.22	0.35
	1ej0A_74	0.34	0.21	0.16	0.16	0.15	0.18
	1i0hA_123	0.62	0.26	0.22	0.09	0.17	0.17
	1id0A_405	0.67	0.72	0.33	0.20	0.19	0.28
	1qnrA_195	0.49	0.39	0.32	0.23	0.23	0.40
	1qopA_44	0.63	0.61	0.13	0.28	0.30	0.22
	1tca_95	0.39	0.28	0.15	0.08	0.09	0.07
	1thfD_121	0.50	0.36	0.11	0.21	0.21	0.11
	Average	0.56	0.40	0.20	0.20	0.22	0.23
8-res	1cruA_85	1.75	0.99	1.48	0.60	0.62	0.95
	1ctqA_144	1.34	0.96	1.37	0.62	0.56	0.54
	1d8wA_334	1.51	0.37	1.18	0.96	0.78	1.29
	1ds1A_20	1.58	1.30	0.93	0.80	0.73	0.67
	1gk8A_122	1.68	1.29	0.96	0.79	0.62	0.98
	1i0hA_145	1.35	0.36	1.37	0.88	0.74	0.35
	1ixh_106	1.61	2.36	1.21	0.59	0.57	0.38
	1lam_420	1.60	0.83	0.90	0.79	0.66	1.10
	1qopB_14	1.85	0.69	1.24	0.72	0.92	0.83
	3chbD_51	1.66	0.96	1.23	1.03	1.03	0.55
	Average	1.59	1.01	1.19	0.78	0.72	0.76
12-res	1cruA_358	2.54	2.00	2.39	2.27	2.07	2.21
	1ctqA_26	2.49	1.86	2.54	1.72	1.66	1.44
	1d4oA_88	2.33	1.60	2.44	0.84	0.82	2.64
	1d8wA_46	4.83	2.94	2.17	2.11	2.09	1.58
	1ds1A_282	3.04	3.10	2.33	2.16	2.10	1.43
	1dysA_291	2.48	3.04	2.08	1.83	1.67	1.58
	1eguA_508	2.14	2.82	2.36	1.68	1.71	1.43
	1f74A_11	2.72	1.53	2.23	1.33	1.44	1.31
	1qlwA_31	3.38	2.32	1.73	2.11	2.20	2.20
	1qopA_178	4.57	2.18	2.21	2.37	2.36	1.69
	Average	3.05	2.34	2.25	1.84	1.81	1.75

Table 3.3: Average minimum RMSD obtained, for the 89 long loops in Zhao et al. (2011).

Loop Length	14	15	16	17
Number of Loops	36	31	13	9
Average Minimum RMSD	2.17	2.27	2.39	2.62

closure and energy make this an intuitive choice to provide sufficient guidance. At this amino acid level, a follow-up extension might use conditional Ramachandran maps to leverage the dependencies between the torsion angles of adjacent residues.

The general sequential importance sampling method (Liu and Chen, 1998; Liu, 2001; Rosenbluth and Rosenbluth, 1955) has been applied previously to sample whole protein chains (Zhang and Liu, 2002; Gan et al., 2000; Grassberger, 1997; Hsu et al., 2003; Zhang et al., 2004, 2007a,b, 2008; Meirovitch, 1982; Cheluvaraja and Meirovitch, 2004; Mamonov et al., 2011; Lin et al., 2008; Zhang et al., 2009; Lin et al., 2011) and to sample side chains (Zhang and Liu, 2006; Jain et al., 2006). In a MCMC setting, it is also known as configurational bias Monte Carlo (CBMC), originally developed by Siepmann and Frenkel (1992) and Frenkel et al. (1992). In this study, we have built on these ideas to develop an effective fragment closure method useful for protein structure simulation.

The potential energy function used in this study to guide the conformation sampling is effective for generating high quality fragment conformations without the need for a very large amount of samples. As such, loop samples generated by FRESS can provide promising initial conformations for further loop modeling and prediction within a small computational budget. The energy function used in this study for sampling has not been optimized to discriminate natives from decoys, compared to

the more sophisticated energy functions used by other researchers in loop prediction. Good accuracy in loop prediction requires both an efficient sampling method and an accurate energy function. This study focuses on the sampling method for loop modeling and the results are also evaluated for the same purpose. We will undertake the loop prediction problem in a future work.

Bibliography

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, 25, 3389–3402.

Anfinsen, C. B. et al. (1973), "Principles that govern the folding of protein chains," *Science*, 181, 223–230.

Baker, D. and Sali, A. (2001), "Protein structure prediction and structural genomics," *Science Signaling*, 294, 93.

Baragatti, M., Grimaud, A., and Pommeret, D. (2012), "Parallel tempering with equi-energy moves," *Statistics and Computing*, 1–17.

Benkert, P., Tosatto, S. C., and Schomburg, D. (2008), "QMEAN: A comprehensive scoring function for model quality assessment," *Proteins: Structure, Function, and Bioinformatics*, 71, 261–277.

Berger, B. and Leighton, T. (1998), "Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete," *Journal of Computational Biology*, 5, 27–40.

Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. (2000), "The Protein Data Bank," *Nucleic Acids Research*, 28, 235–242.

Betancourt, M. R. (2005), "Efficient Monte Carlo trial moves for polypeptide simulations," *Journal of Chemical Physics*, 123.

Biegler, L., Damiano, J., and Blau, G. (1986), "Nonlinear parameter estimation: a case study comparison," *AIChE Journal*, 32, 29–45.

Bonneau, R. and Baker, D. (2001), "Ab initio protein structure prediction: Progress and prospects," *Annual Review of Biophysics and Biomolecular Structure*, 30, 173–189.

Bruccoleri, R. E. and Karplus, M. (1990), "Conformational sampling using high-temperature molecular dynamics," *Biopolymers*, 29, 1847–1862.

Cahill, S., Cahill, M., and Cahill, K. (2003), "On the kinematics of protein folding," *Journal of Computational Chemistry*, 24, 1364–1370.

Campbell, D. (2007), "Bayesian collocation tempering and generalized profiling for estimation of parameters from differential equation models," Ph.D. thesis, McGill University, Montreal.

Canutescu, A. and Dunbrack, R. (2003), "Cyclic coordinate descent: A robotics algorithm for protein loop closure," *Protein Science*, 12, 963–972.

Canutescu, A. A., Shelenkov, A. A., and Dunbrack, R. L. (2003), "A graph-theory algorithm for rapid protein side-chain prediction," *Protein Science*, 12, 2001–2014.

Cheluvaraja, S. and Meirovitch, H. (2004), "Simulation method for calculating the entropy and free energy of peptides and proteins," *Proceedings of the National Academy of Sciences of the United States of America*, 101, 9241–9246.

Chen, L. and Xu, J.-c. (2004), "Optimal Delaunay Triangulations," *Journal of Computational Mathematics*, 22, 299–308.

Chuang, G.-Y., Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. (2008), "DARS (Decoys As the Reference State) potentials for protein-protein docking," *Biophysical Journal*, 95, 4217–4227.

Collura, V., Higo, J., and Garnier, J. (1993), "Modeling of Protein Loops by Simulated Annealing," *Protein Science*, 2, 1502–1510.

Coutsias, E., Seok, C., Jacobson, M., and Dill, K. (2004), "A kinematic view of loop closure," *Journal of Computational Chemistry*, 25, 510–528.

Cui, M., Mezei, M., and Osman, R. (2008), "Prediction of protein loop structures using a local move Monte Carlo approach and a grid-based force field," *Protein Engineering Design and Selection*, 21, 729–735.

da Silva, R., Degreve, L., and Caliri, A. (2004), "LMProt: An efficient algorithm for Monte Carlo sampling of protein conformational space," *Biophysical Journal*, 87, 1567–1577.

de Bakker, P. I., DePristo, M. A., Burke, D. F., and Blundell, T. L. (2003), "Ab initio construction of polypeptide fragments: Accuracy of loop decoy discrimination by an all-atom statistical potential and the AMBER force field with the Generalized Born solvation model," *Proteins*, 51, 21–40.

Dunbrack Jr, R. L. and Karplus, M. (1993), "Backbone-dependent rotamer library for proteins," *Journal of Molecular Biology*, 230, 543–574.

Engh, R. and Huber, R. (1991), "Accurate Bond and Angle Parameters for X-ray Protein-structure Refinement," *Acta Crystallographica Section A: Foundations*, 47, 392–400.

Eswar, N., Webb, B., Marti-Renom, M. A., Madhusudhan, M., Eramian, D., Shen, M.-y., Pieper, U., and Sali, A. (2006), "Comparative protein structure modeling using Modeller," *Current Protocols in Bioinformatics*, 5–6.

Fiser, A., Do, R. K., and Sali, A. (2000), "Modeling of loops in protein structures." *Protein Science*, 9, 1753–1773.

Frenkel, D., Mooij, G., and Smit, B. (1992), "Novel Scheme to Study Structural and Thermal-properties of Continuously Deformable Molecules," *Journal of Physics: Condensed Matter*, 4, 3053–3076.

Fujitsuka, Y., Takada, S., Luthey-Schulten, Z. A., and Wolynes, P. G. (2004), "Optimizing physical energy functions for protein folding," *Proteins: Structure, Function, and Bioinformatics*, 54, 88–103.

Gan, H., Tropsha, A., and Schlick, T. (2000), "Generating folded protein structures with a lattice growth algorithm," *Journal of Chemical Physics*, 113, 5511–5524.

Gelman, A., Bois, F., and Jiang, J. (1996), "Physiological pharmacokinetic analysis using population modeling and informative prior distributions," *Journal of the American Statistical Association*, 91, 1400–1412.

Go, N. and Scheraga, H. A. (1970), "Ring Closure and Local Conformational Deformations of Chain Molecules," *Macromolecules*, 3, 178–187.

Grassberger, P. (1997), "Pruned-enriched Rosenbluth method: Simulations of theta polymers of chain length up to 1,000,000," *Physical Review E*, 56, 3682–3693.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006), "DRAM: efficient adaptive MCMC," *Statistics and Computing*, 16, 339–354.

Hastie, T. and Tibshirani, R. (1986), "Generalized additive models," *Statistical science*, 297–310.

Hirata, H., Yoshiura, S., Ohtsuka, T., Bessho, Y., Harada, T., Yoshikawa, K., and Kageyama, R. (2002), "Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop," *Science Signaling*, 298, 840.

Hsu, H., Mehra, V., Nadler, W., and Grassberger, P. (2003), "Growth-based optimization algorithm for lattice heteropolymers," *Physical Review E*, 68, 021113.

Huang, Y., Liu, D., and Wu, H. (2006), "Hierarchical Bayesian methods for estimation of parameters in a longitudinal HIV dynamic system," *Biometrics*, 62, 413–423.

Jacobson, M., Pincus, D., Rapp, C., Day, T., Honig, B., Shaw, D., and Friesner, R. (2004), "A hierarchical approach to all-atom protein loop prediction," *Proteins*, 55, 351–367.

Jain, T., Cerutti, D., and McCammon, J. (2006), "Configurational-bias sampling technique for predicting side-chain conformations in proteins." *Protein Science*, 15, 2029–2039.

Kabsch, W. and Sander, C. (1983), "Dictionary of Protein Secondary Structure - Pattern-recognition of Hydrogen-Bonded and Geometrical Features," *Biopolymers*, 22, 2577–2637.

Kaminski, G., Friesner, R., Tirado-Rives, J., and Jorgensen, W. (2001), "Evaluation and reparametrization of the OPLS-AA force field for proteins via comparison with accurate quantum chemical calculations on Peptides." *Journal of Physical Chemistry B*, 105, 6474–6487.

Karplus, P. (1996), "Experimentally observed conformation-dependent geometry and hidden strain in proteins." *Protein Science*, 5, 1406–1420.

Kim, D. E., Blum, B., Bradley, P., and Baker, D. (2009), "Sampling Bottlenecks in De novo Protein Structure Prediction," *Journal of Molecular Biology*, 393, 249–260.

Koehl, P. and Delarue, M. (1995), "A self consistent mean field approach to simultaneous gap closure and side-chain positioning in homology modelling," *Nature Structural Biology*, 2, 163–170.

Kong, A. (1992), "A note on importance sampling using standardized weights," Tech. Rep. 348, University of Chicago, Dept. of Statistics.

Kou, S., Zhou, Q., and Wong, W. H. (2006), "Discussion paper equi-energy sampler with applications in statistical inference and statistical mechanics," *The Annals of Statistics*, 1581–1619.

Krivov, G. G., Shapovalov, M. V., and Dunbrack, R. L. (2009), "Improved prediction of protein side-chain conformations with SCWRL4," *Proteins: Structure, Function, and Bioinformatics*, 77, 778–795.

Kryshtafovych, A., Venclovas, Č., Fidelis, K., and Moult, J. (2005), "Progress over the first decade of CASP experiments," *Proteins: Structure, Function, and Bioinformatics*, 61, 225–236.

Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., and Baker, D. (2003), "Design of a novel globular protein fold with atomic-level accuracy," *Science*, 302, 1364–1368.

Lee, A., Streinu, I., and Brock, O. (2005), "A methodology for efficiently sampling the conformation space of molecular structures," *Physical Biology*, 2, S108–S115.

Lee, J., Lee, D., Park, H., Coutsias, E. A., and Seok, C. (2010), "Protein loop modeling by using fragment assembly and analytical loop closure," *Proteins*, 78, 3428–3436.

Liang, S., Zhou, Y., Grishin, N., and Standley, D. M. (2011), "Protein side chain modeling with orientation-dependent atomic force fields derived by series expansions," *Journal of Computational Chemistry*, 32, 1680–1686.

Lin, M., Lu, H.-M., Chen, R., and Liang, J. (2008), "Generating properly weighted ensemble of conformations of proteins from sparse or indirect distance constraints," *Journal of Chemical Physics*, 129.

Lin, M., Zhang, J., Lu, H.-M., Chen, R., and Liang, J. (2011), "Constrained proper sampling of conformations of transition state ensemble of protein folding," *Journal of Chemical Physics*, 134.

Liu, J. and Chen, R. (1998), "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, 93, 1032–1044.

Liu, J. S. (2001), *Monte Carlo strategies in scientific computing*, Springer-Verlag.

Liu, P., Zhu, F., Rassokhin, D. N., and Agrafiotis, D. K. (2009), "A self-organizing algorithm for modeling protein loops." *PLoS Computational Biology*, 5, e1000478.

Liu, Z., Mao, F., Li, W., Han, Y., and Lai, L. (2000), "Calculation of protein surface loops using Monte-Carlo simulated annealing simulation," *Journal of Molecular Modeling*, 6, 1–8.

Lu, G. Y. and Wong, D. W. (2008), "An adaptive inverse-distance weighting spatial interpolation technique," *Computers & Geosciences*, 34, 1044–1055.

MacCallum, J. L., Pérez, A., Schnieders, M. J., Hua, L., Jacobson, M. P., and Dill, K. A. (2011), "Assessment of protein structure refinement in CASP9," *Proteins: Structure, Function, and Bioinformatics*, 79, 74–90.

Mamonov, A. B., Zhang, X., and Zuckerman, D. M. (2011), "Rapid Sampling of All-Atom Peptides Using a Library-Based Polymer-Growth Approach," *Journal of Computational Chemistry*, 32, 396–405.

Mandell, D. J., Coutsias, E. A., and Kortemme, T. (2009), "Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling." *Nature Methods*, 6, 551–552.

Martí-Renom, M. A., Stuart, A. C., Fiser, A., Sánchez, R., Melo, F., and Šali, A. (2000), "Comparative protein structure modeling of genes and genomes," *Annual Review of Biophysics and Biomolecular Structure*, 29, 291–325.

Meirovitch, H. (1982), "A new method for simulation of real chains: Scanning future steps," *Journal of Physics A: Mathematical and General*, 15, L735–L741.

Miao, H., Dykes, C., Demeter, L. M., Cavenaugh, J., Park, S. Y., Perelson, A. S., and Wu, H. (2008), "Modeling and estimation of kinetic parameters and replicative fitness of HIV-1 from flow-cytometry-based growth competition experiments," *Bulletin of Mathematical Biology*, 70, 1749–1771.

Miao, H., Dykes, C., Demeter, L. M., and Wu, H. (2009), "Differential equation modeling of HIV viral fitness experiments: model identification, model selection, and multimodel inference," *Biometrics*, 65, 292–300.

Moennigmann, M. and Floudas, C. A. (2005), "Protein loop structure prediction with flexible stem geometries." *Proteins*, 61, 748–762.

Moré, J. J. (1978), "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*, Springer, pp. 105–116.

Moult, J., Fidelis, K., Kryshtafovych, A., Rost, B., and Tramontano, A. (2009), "Critical assessment of methods of protein structure predictionround VIII," *Proteins: Structure, Function, and Bioinformatics*, 77, 1–4.

Noonan, K., O'Brien, D., and Snoeyink, J. (2005), "Probik: Protein backbone motion by inverse kinematics," *International Journal of Robotics Research*, 24, 971–982.

Omohundro, S. M. (1989), *The Delaunay triangulation and function learning*, International Computer Science Institute.

Peng, H.-P. and Yang, A.-S. (2007), "Modeling protein loops with knowledge-based prediction of sequence-structure alignment." *Bioinformatics*, 23, 2836–2842.

Qian, B., Raman, S., Das, R., Bradley, P., McCoy, A. J., Read, R. J., and Baker, D. (2007), "High-resolution structure prediction and the crystallographic phase problem," *Nature*, 450, 259–264.

Ramachandran, G., Ramakrishnan, C., and Saisekharan, V. (1963), "Stereochemistry of Polypeptide Chain Configurations," *Journal of Molecular Biology*, 7, 95–99.

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007), "Parameter estimation for differential equations: a generalized smoothing approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69, 741–796.

Rohl, C. A., Strauss, C. E., Misura, K. M., Baker, D., et al. (2004a), "Protein structure prediction using Rosetta." *Methods in Enzymology*, 383, 66.

Rohl, C. A., Strauss, C. E. M., Chivian, D., and Baker, D. (2004b), "Modeling structurally variable regions in homologous proteins with rosetta." *Proteins*, 55, 656–677.

Rosenbluth, M. and Rosenbluth, A. (1955), "Monte Carlo simulations of the average extension of molecular chains," *Journal of Chemical Physics*, 23, 356–359.

Roy, A., Kucukural, A., and Zhang, Y. (2010), "I-TASSER: a unified platform for automated protein structure and function prediction," *Nature Protocols*, 5, 725–738.

Sellers, B. D., Zhu, K., Zhao, S., Friesner, R. A., and Jacobson, M. P. (2008), "Toward better refinement of comparative models: predicting loops in inexact environments." *Proteins*, 72, 959–971.

Shen, M.-y. and Sali, A. (2006), "Statistical potential for assessment and prediction of protein structures," *Protein Science*, 15, 2507–2524.

Shenkin, P., Yarmush, D., Fine, R., Wang, H., and Levinthal, C. (1987), "Predicting Antibody Hypervariable Loop Conformation .1. Ensembles of Random Conformations for Ring-like Structures," *Biopolymers*, 26, 2053–2085.

Siepmann, J. and Frenkel, D. (1992), "Configurational Bias Monte-Carlo - A New Sampling Scheme for Flexible Chains," *Molecular Physics*, 75, 59–70.

Simons, K. T., Bonneau, R., Ruczinski, I., and Baker, D. (1999a), "Ab initio protein structure prediction of CASP III targets using ROSETTA," *Proteins: Structure, Function, and Bioinformatics*, 37, 171–176.

Simons, K. T., Ruczinski, I., Kooperberg, C., Fox, B. A., Bystroff, C., and Baker, D. (1999b), "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins," *Proteins: Structure, Function, and Bioinformatics*, 34, 82–95.

Slabinski, L., Jaroszewski, L., Rodrigues, A. P., Rychlewski, L., Wilson, I. A., Lesley, S. A., and Godzik, A. (2007), "The challenge of protein structure determination-lessons from structural genomics," *Protein Science*, 16, 2472–2482.

Söding, J., Biegert, A., and Lupas, A. N. (2005), "The HHpred interactive server for protein homology detection and structure prediction," *Nucleic Acids Research*, 33, W244–W248.

Soto, C. S., Fasnacht, M., Zhu, J., Forrest, L., and Honig, B. (2008), "Loop modeling: Sampling, filtering, and scoring," *Proteins*, 70, 834–843.

Swendsen, R. H. and Wang, J.-S. (1986), "Replica Monte Carlo simulation of spin-glasses," *Physical Review Letters*, 57, 2607–2609.

Tsai, J., Bonneau, R., Morozov, A. V., Kuhlman, B., Rohl, C. A., and Baker, D. (2003), "An improved protein decoy set for testing energy functions for protein structure prediction," *Proteins: Structure, Function, and Bioinformatics*, 53, 76–87.

Tuffery, P., Etchebest, C., Hazout, S., and Lavery, R. (1991), "A new approach to the rapid determination of protein side chain conformations," *Journal of Biomolecular Structure and Dynamics*, 8, 1267–1289.

Uhlherr, A. (2000), "Monte Carlo conformational sampling of the internal degrees of freedom of chain molecules," *Macromolecules*, 33, 1351–1360.

Uhlherr, A., Mavrantzas, V., Doxastakis, M., and Theodorou, D. (2001), "Directed bridging methods for fast atomistic Monte Carlo simulations of bulk polymers," *Macromolecules*, 34, 8554–8568.

Vendruscolo, M. (1997), "Modified configurational bias monte carlo method for simulation of polymer systems," *Journal of Chemical Physics*, 106, 2970–2976.

Wedemeyer, W. and Scheraga, H. (1999), "Exact analytical loop closure in proteins using polynomial equations," *Journal of Computational Chemistry*, 20, 819–844.

Wick, C. and Siepmann, J. (2000), "Self-adapting fixed-end-point configurational-bias Monte Carlo method for the regrowth of interior segments of chain molecules with strong intramolecular interactions," *Macromolecules*, 33, 7207–7218.

Wolynes, P. G. (2005), "Energy landscapes and solved protein–folding problems," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 363, 453–467.

Wong, W., Cui, Y., and Chen, R. (1998), "Torsional relaxation for biopolymers." *Journal of Computational Biology*, 5, 655–665.

Wood, S. N. (2006a), *Generalized additive models: an introduction with R*, vol. 66, Chapman & Hall.

— (2006b), "Low-Rank Scale-Invariant Tensor Product Smooths for Generalized Additive Mixed Models," *Biometrics*, 62, 1025–1036.

Xiang, Z. and Honig, B. (2001), "Extending the accuracy limits of prediction for side-chain conformations," *Journal of Molecular Biology*, 311, 421–430.

Xiang, Z., Soto, C., and Honig, B. (2002), "Evaluating conformational free energies: The colony energy and its application to the problem of loop prediction," *Proceedings of the National Academy of Sciences of the United States of America*, 99, 7432–7437.

Yang, Y. and Zhou, Y. (2008), "Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions," *Protein Science*, 17, 1212–1219.

Zemla, A. (2003), "LGA: a method for finding 3D similarities in protein structures," *Nucleic Acids Research*, 31, 3370–3374.

Zhang, C., Liu, S., and Zhou, Y. (2004), "Accurate and efficient loop selections by the DFIRE-based all-atom statistical potential," *Protein Science*, 13, 391–399.

Zhang, J., Chen, Y., Chen, R., and Liang, J. (2004), "Importance of chirality and reduced flexibility of protein side chains: A study with square and tetrahedral lattice models," *Journal of Chemical Physics*, 121, 592–603.

Zhang, J., Dundas, J., Lin, M., Chen, R., Wang, W., and Liang, J. (2009), "Prediction of geometrically feasible three-dimensional structures of pseudoknotted RNA through free energy estimation," *RNA*, 15, 2248–2263.

Zhang, J., Kou, S. C., and Liu, J. S. (2007a), "Biopolymer structure simulation and optimization via fragment regrowth Monte Carlo," *Journal of Chemical Physics*, 126.

Zhang, J., Lin, M., Chen, R., Liang, J., and Liu, J. S. (2007b), "Monte Carlo sampling of near-native structures of proteins with applications," *Proteins*, 66, 61–68.

Zhang, J., Lin, M., Chen, R., Wang, W., and Liang, J. (2008), "Discrete state model and accurate estimation of loop entropy of RNA secondary structures," *Journal of Chemical Physics*, 128.

Zhang, J. and Liu, J. (2002), "A new sequential importance sampling method and its application to the two-dimensional hydrophobic-hydrophilic model," *Journal of Chemical Physics*, 117, 3492–3498.

Zhang, J. and Liu, J. S. (2006), "On side-chain conformational entropy of proteins," *PLoS Computational Biology*, 2, 1586–1591.

Zhang, J. and Zhang, Y. (2010), "A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction," *PLoS One*, 5, e15386.

Zhang, Y., Kihara, D., and Skolnick, J. (2002), "Local energy landscape flattening: parallel hyperbolic Monte Carlo sampling of protein folding," *Proteins: Structure, Function, and Bioinformatics*, 48, 192–201.

Zhao, S., Zhu, K., Li, J., and Friesner, R. A. (2011), "Progress in super long loop prediction," *Proteins*, 79, 2920–2935.

Zhou, H. and Zhou, Y. (2002), "Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction," *Protein Science*, 11, 2714–2726.

Zhu, K., Pincus, D. L., Zhao, S., and Friesner, R. A. (2006), "Long loop prediction using the protein local optimization program." *Proteins*, 65, 438–452.