Acknowledgements

I would like to thank the following people

- Jan Jensen
- Casper Steinmann
- More people

Licensing

Dansk Resumé

Publication list

List of publications:

1. Anders S. Christensen, Stephan P. A. Sauer, Jan H. Jensen (2011) Definitive benchmark study of ring current effects on amide proton chemical shifts. *Journal of Chemical Theory and Computation*, 7:2078-2084.

2. Wouter Boomsma, Jes Frellsen, Tim Harder, Sandro Bottaro, Kristoffer E. Johansson, Pengfei Tian, Kasper Stovgaard, Christian Andreetta, Simon Olsson, Jan B. Valentin, Lubomir D. Antonov, Anders S. Christensen, Mikael Borg, Jan H. Jensen, Kresten Lindorff-Larsen, Jesper Ferkinghoff-Borg, Thomas Hamelryck (2013) PHAISTOS: A framework for Markov chain Monte Carlo simulation and inference of protein structure. *Journal of Computational Chemistry*, 34:1697-1705.

3. Anders S. Christensen, Troels E. Linnet, Mikael Borg, Wouter Boomsma, Kresten Lindorff-Larsen, Thomas Hamelryck, Jan H. Jensen (2013) Protein Structure Validation and Refinement Using Amide Proton Chemical Shifts Derived from Quantum Mechanics. *PLoS ONE* 8:e84123.

4. Anders S. Christensen, Thomas Hamelryck, Jan H. Jensen (2014) FragBuilder: An efficient Python library to setup quantum chemistry calculations on peptides models. *PeerJ* (accepted).

5. Anders S. Christensen, Lars Bratholm, Simon Olsson, Thomas Hamelryck, Jan H. Jensen (2014) Weighting of chemical shift evidence in Monte Carlo simulation of proteins. *ShareLatex* (unpublished).

6. Torus-DBN-CS-LARS

7. J-coupling Casper/Kongen

List of public code:

1. FragBuilder (BSD license) `https://github.com/jensengroup/fragbuilder`

2. CamShift module (BSD license) `https://github.com/jensengroup/camshift-phaistos`

3. ProCS module (BSD license) `https://github.com/jensengroup/procs-phaistos`

4. PHAISTOS (GPL license) `https://svn.code.sf.net/p/phaistos/code/trunk`

5. GAMESS patch FMO-RHF:MP2 (GAMESS license/free) `https://github.com/andersx/fmo-rhf-mp2`

6. PHAISTOS GUI (BSD license) `https://github.com/andersx/guistos`

List of other publications:

1. Casper Steinmann, Kristoffer L. Blædel, Anders S. Christensen, Jan H. Jensen (2013) Interface of the polarizable continuum model of solvation with semi-empirical methods in the GAMESS program. *PLoS ONE* 8:e67725.

2. Anders S. Christensen, Casper Steinmann, Dmitri G. Fedorov, Jan H. Jensen (2013) Hybrid RHF/MP2 geometry optimizations with the Effective Fragment Molecular Orbital Method. *PLoS ONE* (accepted).

3. HF-3c Jimmy

4. PM6 Jimmy

5. h-bond Jimmy

Contents

Acknowledgement i

Dansk Resumé (Danish Summary) ii

Publication list iii

1 Introduction 2
1.1 Computational methods . . . 2

2 Introduction to PHAISTOS 4
2.1 Markov Chain Monte Carlo . . . 4
2.1.1 Metropolis-Hastings . . . 4
2.1.2 Generalized Ensembles . . . 5
2.2 Monte Carlo Moves Using Generative Probabilistic Models . . . 5
2.3 Note on protein Folding . . . 8

3 Chemical shifts in a probabilistic framework 9
3.1 Hybrid energy schemes . . . 9
3.2 Defining an energy function from Bayes' theorem . . . 10
3.2.1 Gaussian error model . . . 11
3.2.2 Cauchy error model . . . 13
3.2.3 Marginalization of Weighting parameter . . . 14
3.2.4 Soft Square-Well Energy Function . . . 14
3.3 Sampling strategy for weight parameters . . . 15
3.3.1 Molecular mechanics force field . . . 15
3.4 Results . . . 15
3.4.1 Results – sampling of weight parameters . . . 15
3.4.2 Performance of energy functions . . . 16

4 Graphical User Interface for PHAISTOS 20

5 Structure Based Prediction of Protein Chemical Shifts 23

6 Determined protein structures 24
6.1 Barley Chymotrypsin Inhibitor II . . . 24
6.1.1 Computational methodology . . . 24
6.1.2 Folding results . . . 24
6.2 Folding of small proteins (<100 AA) . . . 25
6.3 Folding of larger proteins (>100 AA) . . . 26
6.3.1 Folding protocol . . . 26
6.3.2 Rhodopsin (225 residues) . . . 27
6.4 Evolutionary distance constraints . . . 29

Chapter 1

Introduction

The chemistry of a protein is tightly linked to its 3-dimensional structure. For this reason, protein structure determination is the basis of rational understanding of the chemistry of biological processes involving proteins.

Most currently known protein structures have been solved by X-ray crystallography. One requirement for solving a structure this way is that the protein will crystallize. Modern crystallization methods, however, only have a success rate of 5% [REF XX]. In these cases, nucleic magnetic resonance (NMR) methods may be used with some success. Currently the Protein Data Bank [REF XX] contain 90,000 structures solved by X-ray and 9,000 structures solved by NMR methods, and around 10,000 X-ray and 500 NMR structures are being submitted each year. [REF X] `http://www.pdb.org/pdb/static.do?p=general_information/pdb_statistics/index.html`

Conventional NMR protein structure determination methods records a multidimensional spectrum that correlate the resonance frequencies of several nuclei at the same time. From this spectrum, the common work flow is to first assign the chemical shifts of each nuclei. This process is largely automated for backbone nuclei, but is more involved for side chain atoms. This assignment information is used to identify peaks in the spectrum which correspond to distance restraints (NOE restraints) between pairs of atoms. These distance restraints are the used to generate ensembles of structures that satisfy the given set of restraints.

Protein NMR spectroscopy, however, has several limitations. Large proteins have very crowded spectra, which complicates assignment, due to broad peaks and resulting spectral overlap. This is a substantial hindrance to assignment of the chemical shifts and the valuable NOE restraints. Consequently, around 95% of all NMR structures in the PDB database thus have a size of only 200 amino acids or less. This can be compared to the average sizes of proteins in humans and *E. coli*, which are around 400-600 and 200-400, respectively. These problem can be somewhat alleviate by deuteration which, however, decreases to number NOE restraints that can be obtained. Isotope labeling schemes which selectively label only certain side chains have been invented, as an efficient strategy for such problems.

1.1 Computational methods

A different approach to solving a protein structure from the amino acid sequence is simulation of the energy landscape of the protein. This pratice is also refered to as *protein folding*. In this approach, the possible conformations are sampled using a description of the physics of the proteins. Such *ab initio* approaches have been used to determine structures up to XX using Monte Carlo simulations the ROSETTA program [REF XX]. Another notable example is the simultaneous determination of structure and dynamics of several small proteins via very long molecular dynamics (MD) simulations using the Anton computer [REF XX].

While these methods do not require any experimental input, the are extremely demanding in

terms of the computational resources that are required. Furthermore, they usually fail to converge for structures > 100 amino acids.

The ROSETTA methodology is (currently) arguably the most successful method to determine a protein structure computationally. Recently, the Baker group showed, that inclusion of backbone chemical shifts and RDC data vastly improved the ROSETTA protocol and allowed structures up to 150 residues to be determined. The basis of ROSETTA fragment-assembly of local protein structure, combined with refinement using an energy function that has been demonstrated to work remarkably. Briefly described, the all-atom ROSETTA energy function consists of several additive terms such as Lennard-Jones potentials, terms for solvent exposure, hydrogen bonding, electrostatic pair-interactions and dispersion interactions, and finally torsional potentials for backbone and side chain angles. The demonstrated accuracy of the energy function does come at the cost of computational speed and incomplete conformational sampling seems to be the prohibitive for further success for ROSETTA. This protocol has recently been further improved with inclusion of very sparse NOE data.

This allowed 7 structures around 200 amino acids to be determined, to an accuracy of between 2.5 and 3.9 Å from the corresponding experimental X-ray structures Furthermore, a good structure for the 376 amino acids maltose binding protein could even be determined, but this required substantially more NOE data. These simulations, however required a 512-cores super computer for running several days, for each protein.

Another notable example of protein structure determination methods that employ NMR data is is the CHESHIRE method [REF XX]. The CHESHIRE method was the first method which solved structures using only chemical shifts, and uses a fragment-assembly approach followed by a Monte Carlo refinement using an all-atom force-field and an energy function that includes chemical shifts. This method was used to determine the protein structures from chemical shifts, and was demonstrated on 11 proteins between 54 and 123 amino acids in size to an accuracy of around 1.5 Å from the corresponding experimental X-ray structures.

In the following section, the PHAISTOS program is introduced, and the formalism for inclusion of chemical shifts in PHAISTOS is derived. This is an attempt to address the two central challenges in protein folding: (1) complete conformational sampling and (2) accurate energy scoring of conformational samples.

These challenges are met as follows: (1) using a recently developed biased conformational sampling method and (2) by parametrizing an accurate chemical shift predictor and deriving an energy function based rigorously on Bayesian statistics, which allows this to be combined with existing energy functions in PHAISTOS.

The approach will be demonstrated on a test-set of protein with known structures ranging from 55 to 269 residues.

Chapter 2

Introduction to PHAISTOS

This section servers as an introduction to the PHAISTOS program, and a (very) brief introduction to the theory behind PHAISTOS. This will give the relevant background to read the next chapters.

2.1 Markov Chain Monte Carlo

One of the primary goals of simulations in PHAISTOS is to construct the Boltzmann distribution of a protein via Markov chain Monte Carlo (MCMC) sampling for a given potential energy surface at a given temperature. The Boltzmann distribution of a protein structure, $\mathbf{X}$, at a given temperature, T, is given by:

$$p(\mathbf{X}) = \frac{1}{Z(T)} \exp\left(\frac{-E}{k_{\mathrm{B}}T}\right), \tag{2.1}$$

where $k_{\mathrm{B}}T$ is Boltzmann's constant and $Z(T)$ is the partition function at the given temperature.

In Markov chain Monte Carlo the target distribution obtained by repeatedly proposing updates to the current state, and accepting or rejecting these updates with a certain acceptance probability.

It can be shown, that in for the infinitely sampled distribution to converge to the target distribution, i.e. $p_\infty(\mathbf{X}) = p(\mathbf{X})$, the Monte Carlo moves that are used to propose updates satisfy the principle of detailed balance. That is, the transition from the current state $\mathbf{X}$ to the proposed new state $\mathbf{X}'$ fulfills:

$$p(\mathbf{X})p(\mathbf{X} \to \mathbf{X}') = p(\mathbf{X}')p(\mathbf{X}' \to \mathbf{X}) \tag{2.2}$$

where $p(\mathbf{X} \to \mathbf{X}')$ is the probability to of moving from the state $\mathbf{X}$ to $\mathbf{X}'$ using a given move. If we further factorize $p(\mathbf{X} \to \mathbf{X}')$ into an acceptance probability p_a and a move transition probability p_m, Eqn. 2.2 gives:

$$\frac{p_a(\mathbf{X} \to \mathbf{X}')}{p_a(\mathbf{X}' \to \mathbf{X})} = \frac{p(\mathbf{X}')}{p(\mathbf{X})} \frac{p_m(\mathbf{X}' \to \mathbf{X})}{p_m(\mathbf{X} \to \mathbf{X}')} \tag{2.3}$$

Most of the moves in PHAISTOS are symmetric, that is the move bias ratio $p_m(\mathbf{X}' \to \mathbf{X})/p_m(\mathbf{X} \to \mathbf{X}') = 1$, but for some moves this is not true. These moves can be exploited to vastly speed up convergence or bias the simulation, and are discussed later in Section 2.2.

2.1.1 Metropolis-Hastings

The simplest Monte Carlo method that satisfies Eqn. 2.3 is the Metropolis-Hastings method. Here a transition $\mathbf{X} \to \mathbf{X}'$ is accepted using the Metropolis-Hastings acceptance criterion:

$$p_a(\mathbf{X} \to \mathbf{X}') = \min\left(1, \frac{p(\mathbf{X}')}{p(\mathbf{X})} \frac{p_m(\mathbf{X}' \to \mathbf{X})}{p_m(\mathbf{X} \to \mathbf{X}')}\right) \tag{2.4}$$

Note how the term $1/Z(T)$ term from Eqn. 2.1 cancels out. Evaluation of the partition function is thus not necessary. The Metropolis-Hastings method is efficient when exploring native states, and simulations near the critical temperature. Unfortunately the Metropolis-Hastings method, compared to other MC methods, often gets stuck in local minima, and is therefore generally inefficient when simulating protein folding from an extended strand.

2.1.2 Generalized Ensembles

To avoid the slow convergence problem more advanced MC methods are available in PHAISTOS, which emphasize sampling at low energies, which is generally of higher interest in protein structure determination. These "generalized ensemble" methods are very similar to the Metropolis-Hastings methods, and the main difference in the acceptance criterion is that the target distribution $p(\mathbf{X})$ has been replace by a generalized weight function $w(\mathbf{X})$. The acceptance criterion then becomes:

$$p_a(\mathbf{X} \to \mathbf{X}') = \min\left(1, \frac{w(\mathbf{X}')}{w(\mathbf{X})} \frac{p_m(\mathbf{X}' \to \mathbf{X})}{p_m(\mathbf{X} \to \mathbf{X}')}\right) \tag{2.5}$$

Through reweighting, samples from a converged simulation in a generalized ensemble can be reweighted to correspond to the Boltzmann distribution at a given temperature.

PHAISTOS offer two generalized ensemble methods. In the multicanonical ensemble method, the weight function is $w_{muca}(\mathbf{X}) = 1/g(E(\mathbf{X}))$, where $E(\mathbf{X})$ is the energy of the structure $\mathbf{X}$ and g is the associated density of states. In the inverse-k ensemble, the weight function is given by $w_{1/k}(\mathbf{X}) = 1/k(E(\mathbf{X}))$ where $k(E(\mathbf{X})) = \int_{-\infty}^{E(\mathbf{X})} g(E')dE'$. The since the density of states is generally unknown, the weight-function is estimated during the simulation. PHAISTOS uses the MUNINN library to collect histograms of the energy and efficiently provide an estimate of $w(\mathbf{X})$ on-the-fly.

2.2 Monte Carlo Moves Using Generative Probabilistic Models

PHAISTOS proposes new structure samples using a weighted set of difference MC moves, which each randomly changes the current protein structure in a certain way. Briefly, these are divided in side chain moves and backbone moves. Side chain moves update the rotamer-conformation of a amino-acid single side chain by rotating the dihedral angles on the side chain. Backbone moves either perform a local perturbation to a strand of a only a few amino acids, or rotates a dihedral angle on the backbone.

Using random moves which re-sample angles from a uniform distribution, and then constructing a target distribution via an acceptance criterion is a perfectly valid strategy. However, sampling from a uniform distribution usually lead to slow convergence. A common approach to alleviate this problem is using fragment assembly, in which small fragments of peptides are assembled from a library of common fragment motifs, such as beta-strands, helices and loops. This approach, however, introduces a bias in the selection probability P_s, which must be divided out if the simulation has to obey detailed balance. Furthermore, it is not clear, how to evaluate the move bias ratio $p_m(\mathbf{X}' \to \mathbf{X})/p_m(\mathbf{X} \to \mathbf{X}')$ when sampling from a fragment library.

A related approach to obtain a similar speed up is biased sampling. PHAISTOS supports sampling of both side chain and backbone angles from such generative probabilistic models. In this approach, angles are sampled from distributions that are conditioned on prior knowledge. Two all-atom generative probabilistic models are supported in PHAISTOS. TorusDBN which is a hidden-Markov model of backbone angles, and BASILISK which is a similar model of side chain rotamer-conformations. Both work are continuous models in torsion-angle space. The model that is used in this work is TorusDBN, which is is a model that samples backbone dihedral angles

conditioned on the amino acid sequence. This effectively speeds up convergence of sampling, since uninteresting parts of conformational space in only sampled very rarely. The importance of the TorusDBN model is discussed in chapter [RESULTS XX].

Using models such as TorusDBN and BASILISK introduces a move bias, which compensated for in Eqn. 2.3 by multiplying by the ratio $p_m(\mathbf{X}' \to \mathbf{X})/p_m(\mathbf{X} \to \mathbf{X}')$. It is possible to determine this ration, because the likelihood of sampled values can be calculated in the TorusDBN model. It is thus possible, to recover the target distribution (e.g. the Boltzmann distribution or a generalized ensemble), despite using only biased moves.

Effectively, this turns the target distribution into an effective target distribution. For sampling from the Boltzmann distribution (e.g. using a molecular mechanics force field), the effective target distribution becomes

$$p_e(\mathbf{X}) = p(\mathbf{X})p_m(\mathbf{X}|I), \tag{2.6}$$

where $p_m(\mathbf{X}|I)$ is the probability distribution from the generative model, conditioned on the prior information I available to the model. This is approach is formally equivalent to adding the term $\ln(p_m(\mathbf{X}|I))$ to the physical energy (although this term does not scale with the temperature):

$$\begin{aligned} p_e(\mathbf{X}) &= p(\mathbf{X})p_m(\mathbf{X}|I) \\ &\propto \exp\left(\frac{-E(\mathbf{X})}{k_\mathrm{B}T}\right)p_m(\mathbf{X}|I) \\ &\propto \exp\left(\frac{-E(\mathbf{X})}{k_\mathrm{B}T} - \ln\left(p_m(\mathbf{X}|I)\right)\right) \end{aligned} \tag{2.7}$$

In other words, biased sampling can be regarded as simply use of a better force field, while the convergence of the simulation is vastly improved.

TorusDBN is implemented in two versions; standard TorusDBN which, in brief, is conditioned on only the amino-acid sequence, and TorusDBN-CS which is furthermore based on backbone and beta-carbon chemical shifts. The default TorusDBN model is trained on a set of 1,447 proteins of 180 different SCOP-fold classifications. The default TorusDBN-CS model is trained on 1349 proteins and corresponding chemical shifts from the RefDB training set.

Effectively, proposing structures from TorusDBN biases the simulation towards likely angles within the Ramachandran-plot, and furthermore also towards a certain secondary structure type that is likely for the particular amino acid sequence. The effect of TorusDBN-CS is similar, but the effect is much more pronounced.

Fig. 2.1 shows an example of three different, but typical cases from Ubiquitin. These are alpha-helix, beta-sheet and loop regions. Residue 29 (lysine) is in a typical alpha helix and this corresponds to the most often sampled cluster from both TorusDBN and TorusDBN-CS. TorusDBN-CS, however, very precisely locates the center of the cluster to within around ± 15 degrees. TorusDBN, in contrast, has some sampling density in the regions typical for beta-sheet and left-handed alpha-helices.

For residue 44 (isoleucine) which is in a typical alpha-helix region of Ubiquitin, TorusDBN-CS accurately pinpoints the distribution of samples around the experimental values. TorusDBN, however, manages to rule out left-handed helices, but has a higher sampling density in the alpha-helix region than the beta-sheet region.

The last residue in the examples, residue 60 (aspargine), is located in a loop-region with backbone angles that correspond to a left-handed helix. Both models sample in the correct region, but TorusDBN favor a regular alpha-helix. While TorusDBN-CS heavily favors the correct region, angles that are usually not favored in the Ramachandran plot are also frequently sampled. This is presumably due to less fold-diversity in the training set, compared to the set used to train TorusDBN.

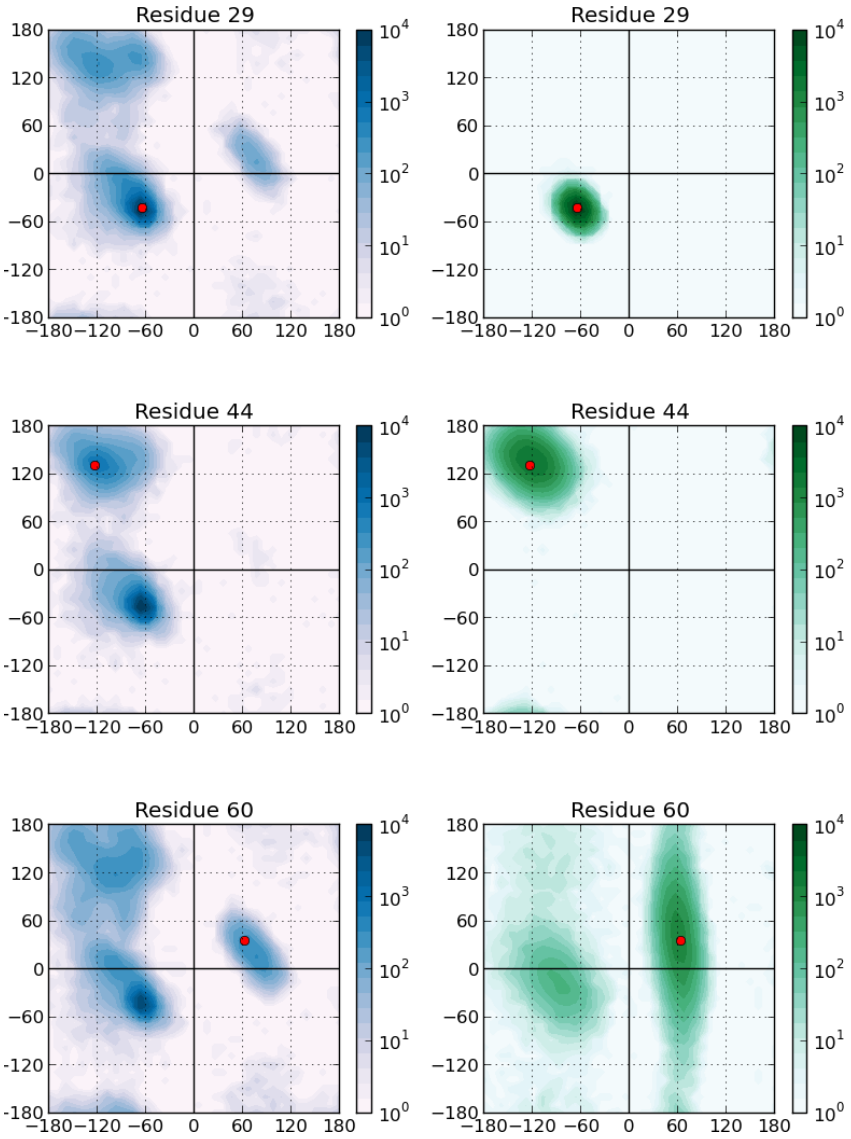


Figure 2.1: Sampling densities from TorusDBN (left/blue) and TorusDBN-CS (right/green) for the residues 29, 44 and 60 in Ubiquitin. Values from the experimental structure 1UBQ are marked with a red dot. Residue 29 (lysine) is located in the middle of an alpha-helix. Residue 44 (isoleucine) is located in a beta-sheet motif, and finally residue 60 (aspargine) is located in a loop region.

2.3 Note on protein Folding

While converged sampling of the potential energy surface will corresponds to the Boltzmann distribution, and the native folded state of the protein will usually correspond to the largest populated cluster of samples.

However, due to practical limits on computational resources, it is generally impossible to perform converged sampling of the potential energy surface. Protein folding simulations from an extended strand will often be very far from convergence.

An often used strategy is to determine a structure by selecting the structure with the lowest energy throughout a simulation. This obviously neglects entropic effect which may be very important in certain cases, but in practice this has shown to be a very efficient strategy to determine structures close to those obtained experimentally. [REF XX]

Chapter 3

Chemical shifts in a probabilistic framework

This section introduces the formalism for Monte Carlo simulations which includes both physical energy terms as well as a probabilistic energy terms based on experimentally observed chemical shifts. The method presented is not new but has not been published in the form presented here.

Working in a probabilistic framework is a powerful strategy for estimation of unknown parameters, and the intention is to present the equations in the form in which they are implemented in PHAISTOS, so that they can easily be re-implemented in other programs by others. Simulations using the CamShift and ProCS chemical shifts predictors presented later in this thesis employ the equations presented in this chapter.

3.1 Hybrid energy schemes

There are several ways to include experimental observations in simulations, and combine these with known laws of physics. A simplistic approach to this problem is to is to define a hybrid energy by defining a penalty function that describes the agreement between experimental data and data calculated from a proposed model with a physical energy (such as from a molecular mechanics force field). A structure can then be determined, for instance, by minimizing

$$E_{\text{hybrid}} = w_{\text{data}}\ E_{\text{data}} + E_{\text{physical}}. \tag{3.1}$$

where w_{data} is the weight that quantifies the belief in the energy-model E_{data} which defines the agreement between the proposed structure and the experimental data relative to the physical energy.

This concept of using a hybrid energy to determine a protein structure was pioneered by Jack and Levitt who simultaneously minimized a molecular mechanics force field energy and the experimental R-factor for the BPTI protein [Jack and Levitt, 1978]. This approach, however, does not uniquely define neither shape nor weight of E_{data}, and the resulting structure will necessarily depend on these (ill-defined) choices.

Consequently, chemical shifts have been combined with physical energies in a multitude of ways, e.g., weighted RMSD values or harmonic constraints. The groups of Bax and Baker added the chi-square agreement between SPARTA predicted chemical shift values and experimental chemical shifts with an empirical weight of 0.25 to the ROSETTA all-atom energy [Shen et al., 2008]. This methodology was used to determine the structure of 16 small to medium sized proteins.

The CHESHIRE method [Cavalli et al., 2007] uses a hybrid energy function, where a classical energy term is divided by the logarithm of a sum of weighted correlation-coefficients between

SHIFTX calculated chemical shifts and experimental values. Here alpha-hydrogen chemical shifts are weighted by a factor of 18 relative nitrogen and carbon chemical shifts which carry a weight of 1. This hybrid energy is used in the refinement step of the CHESHIRE protocol, and was used to determine the structure of 11 proteins to a backbone RMSD of 1.21 to 1.76 Å relative to the corresponding X-ray or NMR structures.

Vendruscolo and co-workers implemented a "square-well soft harmonic potential", and corresponding molecular gradients and were able to run a chemical shift-biased MD simulation using the CamShift chemical shift predictor [Robustelli et al., 2010]. Subsequently, the trajectory snapshots were re-weighted by multiplying the chemical shift energy term by an empirical weight of 5. Using the empirically optimized balance between energy terms, the native state could be determined from the trajectories for 11 small proteins.

In all cases the parameters and weights of E_{data} had to be carefully tweaked by hand, and it is not clear how to choose optimal parameters. For instance, different types of chemical shifts may (for optimal results) require different weighting, and a brute-force optimization of all parameters is not straight-forward.

3.2 Defining an energy function from Bayes' theorem

The inferential structure determination (ISD) principles introduced by Rieping, Habeck and Nigles [Rieping et al., 2005] defines a Bayesian formulation of Eq. 3.1. The ISD approach rigorously defines the shape and weight of the E_{data} term from the definition of an error model, and allows for the weights to be determined automatically as well. In the following section the equations for an ISD approach are derived for combining the knowledge of experimental chemical shifts with a physical energy.

First remember Bayes' theorem which relates a conditional probability (here A given B) with its inverse:

$$p\left(A|B\right) = \frac{p\left(B|A\right)p\left(A\right)}{p\left(B\right)} \tag{3.2}$$

Now consider a set of chemical shifts $\{\delta_i\}$, the weight for each chemical shift restraint $\{w_i\}$ in the simulation, and finally the structure to be determined, $\mathbf{X}$. This introduces an additional parameter, the weights, which must be determined. These weights describe the belief in the model that relates a structure to a chemical shift. In this case, the most likely structure, $\mathbf{X}$, and optimal choice of $\{w_i\}$ given the set of experimental chemical shifts $\{\delta_i\}$ (via Bayes' theorem) can for instance be found by maximizing:

$$\begin{aligned} p\Big(\mathbf{X}, \{w_i\}\Big|\{\delta_i\}\Big) &= \frac{p\Big(\{\delta_i\}\Big|\mathbf{X}, \{w_i\}\Big)p\Big(\mathbf{X}, \{w_i\}\Big)}{p\Big(\{\delta_i\}\Big)} \\ &\propto p\Big(\{\delta_i\}\Big|\mathbf{X}, \{w_i\}\Big)p\Big(\mathbf{X}, \{w_i\}\Big). \end{aligned} \tag{3.3}$$

Here, the *marginal distribution* of $p\left(\{\delta_i\}\right)$ merely serves as a normalizing factor, and can be neglected. The *likelihood* distribution $p\Big(\{\delta_i\}\Big|\mathbf{X}, \{w_i\}\Big)$ describes the likelihood of the experimental chemical shifts, given a structure, $\mathbf{X}$, and the weights $\{w_i\}$. This requires (1) a forward model to calculate chemical shifts from given structure and (2) an error model that relates the degree of belief in the forward model (that is, the weights) to a probability, based on the difference between experimental and calculated values. Later in this chapter, Gaussian and Cauchy distributions are discussed as error models. The forward model here is a chemical shift predictor, e.g. CamShift, ProCS, etc.

If we assume conditional independence, the *prior* $p\Big(\mathbf{X}, \{w_i\}\Big)$ can be separated as

$$p\Big(\mathbf{X}, \{w_i\}\Big) = p\Big(\mathbf{X}\Big)p\Big(\{w_i\}\Big). \tag{3.4}$$

The two priors, $p\Big(\mathbf{X}\Big)$ and $p\Big(\{w_i\}\Big)$, in brief, describe the distribution of *a priori* meaningful structures (i.e. usually the Boltzmann distribution), and the probability distribution of the weights, respectively. In the following $p(\mathbf{X})$ is simply the Boltzmann distribution, i.e.

$$p(\mathbf{X}) = \frac{1}{Z(T)} \exp\left(-\frac{E(\mathbf{X})}{k_{\mathrm{B}}T}\right) \tag{3.5}$$

where $E(\mathbf{X})$ is the (physical) potential energy of the protein structure, most often calculated using a molecular mechanics force field. k_{B} is the Boltzmann constant and T is the temperature of interest. We need not calculate the partition function, $Z(T)$, because the relative energy landscape is invariant under choice of normalization constant. Note that $p(\mathbf{X})$ also can be introduced via conformational sampling from a biased distribution, such as for example TorusDBN or BASILISK (mimicking the Ramachandran plot and side chain rotamer distributions, respectively). This is discussed later in this chapter.

The prior distribution of the weight parameter $p\Big(\{w_i\}\Big)$ is inherently unknown, except that it is some real number. One such *uninformative prior* could for instance be a flat distribution over the positive real line. This distribution, however, may be biased towards very large numbers. A standard method is to use the Jeffreys' prior, which is a generalization of flat priors, and can be used to model such unknown distributions while introducing only minimal bias. In the one parameter case the Jeffrey's prior is given as

$$p(\theta) \propto \sqrt{\mathbf{I}(\theta)}, \tag{3.6}$$

where $\mathbf{I}(\theta)$ is the *Fisher information* defined (in the one parameter case) as

$$\mathbf{I}(\theta) = \left\langle \left(\frac{\partial}{\partial\theta} \ln p(x|\theta) \right)^2 \right\rangle. \tag{3.7}$$

The corresponding priors for the Gaussian and Cauchy distributions are discussed in the next sections.

3.2.1 Gaussian error model

Selecting an error model is the basic assumption that difference (the error) between a chemical shift calculated from a structure and the corresponding experimentally measured chemical shift, given as $\Delta\delta_i(\mathbf{X}) = \left|\delta_i^{\text{predicted}}(\mathbf{X}) - \delta_i^{\text{experimental}}\right|$, is distributed according to some defined distribution. Following the principle of maximum entropy, the Gaussian distribution is the least biasing distribution, and is the least biasing choice of error model. In this case, the weight parameter introduced in the previous section corresponds to the standard deviation, σ of the Gaussian distribution. For simplicity, it is assumed that the mean of the Gaussian is zero. The total likelihood is then the product of the probability of each $\Delta\delta_i(\mathbf{X})$:

$$\begin{aligned} p\Big(\{\delta_i\}\Big|\mathbf{X}, \{\sigma_i\}\Big) &= \prod_{i=0}^{n} p\left(\Delta\delta_i(\mathbf{X})|\sigma_i\right) \\ &\propto \prod_{i=0}^{n} \frac{1}{\sigma_i} \exp\left(-\frac{\Delta\delta_i(\mathbf{X})^2}{2\sigma_i^2}\right) \end{aligned} \tag{3.8}$$

Next we derive Jeffreys' prior for the uncertainty of a generic Gaussian distribution of the form

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)}{2\sigma^2}\right). \tag{3.9}$$

Via Eqn. 3.6, this immediately gives us the Jeffreys' prior:

$$\begin{aligned}
p(\sigma) &\propto \sqrt{\left\langle \left(\frac{\partial}{\partial\sigma} \ln p(x|\mu,\sigma)\right)^2 \right\rangle} \\
&= \sqrt{\left\langle \left(\frac{\partial}{\partial\sigma} \ln\left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)}{2\sigma^2}\right)\right]\right)^2 \right\rangle} \\
&= \sqrt{\left\langle \left(\frac{(x-\mu)-\sigma^2}{\sigma^3}\right)^2 \right\rangle} \\
&= \sqrt{\int_{-\infty}^{\infty} p(x|\mu,\sigma) \left(\frac{(x-\mu)-\sigma^2}{\sigma^3}\right)^2 dx} \\
&= \sqrt{\frac{2}{\sigma^2}} \propto \frac{1}{\sigma}
\end{aligned} \tag{3.10}$$

Practically, it is impossible to have a separate weight for each individual chemical shift, and the chemical shift of nuclei of the same type thus carry the same weight. The forward model is similar for all nuclei of the same type, so this is somewhat well-justified.

In the following equations, j runs over atom types (e.g. C^α or H^α, etc), and i over residue number. Inserting Eqn. 3.8 and Eqn. 3.10 into Eqn. 3.3, we arrive at a total probability of:

$$\begin{aligned}
p\Big(\mathbf{X}, \{\sigma_j\}\Big|\{\delta_{ij}\}\Big) &\propto p\Big(\{\delta_{ij}\}\Big|\mathbf{X}, \{\sigma_j\}\Big) p\Big(\mathbf{X}\Big) p\Big(\{\sigma_j\}\Big) \\
&\propto \prod_{j=0}^{m}\prod_{i=0}^{n} \frac{1}{\sigma_j} \exp\left(-\frac{\Delta\delta_{ij}(\mathbf{X})^2}{2\sigma_j^2}\right) \exp\left(-\frac{E(\mathbf{X})}{k_\mathrm{B}T}\right) \prod_{j=0}^{m} \frac{1}{\sigma_j} \\
&= \prod_{j=0}^{m} \left(\frac{1}{\sigma_j}\right)^n \exp\left(\sum_{i=0}^{n} -\frac{\Delta\delta_{ij}(\mathbf{X})^2}{2\sigma_j^2}\right) \exp\left(-\frac{E(\mathbf{X})}{k_\mathrm{B}T}\right) \prod_{j=0}^{m} \frac{1}{\sigma_j} \\
&= \prod_{j=0}^{m} \left(\frac{1}{\sigma_j}\right)^{n+1} \exp\left(\sum_{i=0}^{n} -\frac{\Delta\delta_{ij}(\mathbf{X})^2}{2\sigma_j^2}\right) \exp\left(-\frac{E(\mathbf{X})}{k_\mathrm{B}T}\right)
\end{aligned} \tag{3.11}$$

This can be converted to the corresponding hybrid-energy:

$$\begin{aligned}
E_\mathrm{hybrid} &= -k_\mathrm{B}T \ln\left(p\Big(\boldsymbol{X}, \{\sigma_i\}\Big|\{\delta_{ij}\}\Big)\right) \\
&= E(\boldsymbol{X}) + k_\mathrm{B}T \sum_{j=0}^{m} (n+1)\ln(\sigma_j) + k_\mathrm{B}T \sum_{j=0}^{m}\sum_{i=0}^{n} \frac{\Delta\delta_{ij}(\mathbf{X})^2}{2\sigma_j^2}
\end{aligned} \tag{3.12}$$

This expression, except for the term $(n+1)\ln(\sigma)$, is essentially an energy function using harmonic constraints. It is, however, the balance between the two terms which include σ that makes things work. The term $(n+1)\ln(\sigma)$ yields the lowest energy for small values of σ, while the term $\frac{\Delta\delta(\mathbf{X})^2}{2\sigma^2}$ is lower for large values of σ.

Furthermore, the effect of the prior is minute: Using Jeffreys' prior this term is $(n+1)\ln(\sigma)$, whereas using a uniform prior the same term is $n\ln(\sigma)$. Since n is the number of measured chemical shifts of a certain type, the value is usually in the order of ~ 100.

3.2.2 Cauchy error model

Due to numerical instabilities in simulation using the Gaussian error model, a similar model was derived, using a Cauchy distribution as error model. The most notable difference between the Gaussian and Cauchy distributions is that the Cauchy distribution has fatter tails, and thus allows for larger outliers. The differences are discussed in further detail in the Results section in this chapter.

Similarly to Eqn. 3.8, we assume that the location parameter of the Caucy-distribution is zero, and use the scale-parameter, γ as the weight. The total likelihood is then:

$$\begin{aligned} p\Big(\{\delta_i\}\Big|\mathbf{X},\{\gamma_i\}\Big) &= \prod_{i=0}^{n} p\left(\Delta\delta_i(\mathbf{X})|\gamma_i\right) \\ &\propto \prod_{i=0}^{n} \frac{1}{\gamma_i\left[1+\left(\frac{\Delta\delta_i(\mathbf{X})}{\gamma_i}\right)^2\right]} \end{aligned} \tag{3.13}$$

And for the γ parameter of the generic Cauchy distribution of the form

$$p(x|x_0,\gamma) = \frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]}, \tag{3.14}$$

we obtain the following Jeffreys' prior:

$$\begin{aligned} p(\gamma) &\propto \sqrt{\left\langle\left(\frac{\partial}{\partial\gamma}\ln p(x|x_0,\gamma)\right)^2\right\rangle} \\ &= \sqrt{\left\langle\left(\frac{\partial}{\partial\gamma}\ln\left[\frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]}\right]\right)^2\right\rangle} \\ &= \sqrt{\left\langle\left(-\frac{\gamma^2-(x-x_0)^2}{\gamma^3+\gamma(x-x_0)^2}\right)^2\right\rangle} \\ &= \sqrt{\int_{-\infty}^{\infty} p(x|x_0,\gamma)\left(-\frac{\gamma^2-(x-x_0)^2}{\gamma^3+\gamma(x-x_0)^2}\right)^2 dx} \\ &= \sqrt{\frac{1}{2\gamma^2}} \propto \frac{1}{\gamma} \end{aligned} \tag{3.15}$$

Again, it is practically impossible to have a separate weight for each individual chemical shift, and the chemical shift of nuclei of the same type thus carry the same weight. In the following equations, j runs over atom types (e.g. C^α or H^α, etc), and i over residue number. Assembling

the Eqn. 3.13 and Eqn. 3.15 into Eqn. 3.3, we arrive at the total probability of:

$$\begin{aligned} p\Big(\mathbf{X},\{\gamma_j\}\Big|\{\delta_{ij}\}\Big) &\propto p\Big(\{\delta_{ij}\}\Big|\mathbf{X},\{\gamma_j\}\Big)p\Big(\mathbf{X},\{\gamma_j\}\Big) \\ &\propto \prod_{j=0}^{m}\prod_{i=0}^{n}\frac{1}{\gamma_j\left[1+\left(\frac{\Delta\delta_{ij}(\mathbf{X})}{\gamma_j}\right)^2\right]}\exp\left(-\frac{E(\mathbf{X})}{k_{\mathrm{B}}T}\right)\prod_{j=0}^{m}\frac{1}{\gamma_j} \\ &= \prod_{j=0}^{m}\left(\frac{1}{\gamma_j}\right)^{n+1}\prod_{i=0}^{n}\frac{1}{1+\left(\frac{\Delta\delta_{ij}(\mathbf{X})}{\gamma_j}\right)^2}\exp\left(-\frac{E(\mathbf{X})}{k_{\mathrm{B}}T}\right) \end{aligned} \tag{3.16}$$

The associated hybrid energy is then given as:

$$\begin{aligned} E_{\mathrm{hybrid}} &= -k_{\mathrm{B}}T\ln\left(p\Big(\boldsymbol{X},\{\gamma_i\}\Big|\{\delta_{ij}\}\Big)\right) \\ &= E(\mathbf{X})+k_{\mathrm{B}}T\sum_{j=0}^{m}(n+1)\ln(\gamma_j)+k_{\mathrm{B}}T\sum_{j=0}^{m}\sum_{i=0}^{n}\ln\left[1+\left(\frac{\Delta\delta_{ij}(\mathbf{X})}{\gamma_j}\right)^2\right] \end{aligned} \tag{3.17}$$

3.2.3 Marginalization of Weighting parameter

A third option also explored here, is the removal of the weight parameter by projection. This procedure is known as *marginalization*, and is carried out by integrating over all values of the weight parameter. While integration is straight-forward for the Gaussian error-model, the similar expression for the Cauchy distribution does not integrate easily, and the Cauchy-model was not investigated here. From the joint probability distribution in Eqn. 3.11 we obtain the following:

$$\begin{aligned} p_{\mathrm{marginal}}\Big(\mathbf{X}\Big|\{\delta_{ij}\}\Big) &= \int_0^\infty p\Big(\{\delta_{ij}\}\Big|\mathbf{X},\{\sigma_j\}\Big)p\Big(\mathbf{X}\Big)p\Big(\{\sigma_j\}\Big)d\sigma \\ &= \int_0^\infty\prod_{j=0}^{m}\left(\frac{1}{\sigma_j}\right)^{n+1}\exp\left(\sum_{i=0}^{n}-\frac{\Delta\delta_{ij}(\mathbf{X})^2}{2\sigma_j^2}\right)\exp\left(-\frac{E(\mathbf{X})}{k_{\mathrm{B}}T}\right)d\sigma \\ &= \prod_{j=0}^{m}\left(\sum_{i=0}^{n}\Delta\delta_{ij}(\mathbf{X})^2\right)^{n/2}\exp\left(-\frac{E(\mathbf{X})}{k_{\mathrm{B}}T}\right) \end{aligned} \tag{3.18}$$

The hybrid energy associated with the marginalized probability is then given as:

$$\begin{aligned} E_{\mathrm{hybrid}} &= -k_{\mathrm{B}}T\ln\left(p_{\mathrm{marginal}}\Big(\boldsymbol{X}\Big|\{\delta_{ij}\}\Big)\right) \\ &= E(\mathbf{X})+\frac{n}{2}\sum_{j=0}^{m}\ln\sum_{i=0}^{n}\Delta\delta_{ij}(\mathbf{X})^2 \end{aligned} \tag{3.19}$$

3.2.4 Soft Square-Well Energy Function

The last type of hybrid energy term explored here, is a potential designed specifially for molecular dynamics simulations biased by the CamShift predictor. [Robustelli et al., 2009, Robustelli et al., 2010] In this case, the hybrid-energy is given as:

$$E_{\mathrm{hybrid}} = E(\mathbf{X})+\alpha E_{\mathrm{CS}}(\mathbf{X},\{\delta_{ij}\}), \tag{3.20}$$

where $E_{\mathrm{CS}}(\mathbf{X},\{\delta_{ij}\})$ is an empirically derived penalty function that has been demonstrated through simulations to work well for protein structure determination. α is a weight parameter

which was set to 1 during simulation. This penalty function is termed a "soft-square harmonic well", and given by:

$$E_{\mathrm{CS}}(\mathbf{X}, \{\delta_{ij}\}) = \sum_{j=0}^{m} \sum_{i=0}^{n} E_{ij}, \tag{3.21}$$

with

$$E_{ij} = \begin{cases} 0 & \text{if } \Delta\delta_{ij}(\mathbf{X}) < n\epsilon_j \\ \left(\frac{\Delta\delta_{ij}(\mathbf{X}) - n\epsilon_j}{\beta_j}\right)^2 & \text{if } n\epsilon_j < \Delta\delta_{ij}(\mathbf{X}) < x_0 \\ \left(\frac{x_0 - n\epsilon_j}{\beta_j}\right)^2 + \gamma \tanh \frac{2(x_0 - n)(\Delta\delta_{ij}(\mathbf{X}) - x_0)}{\gamma\beta_j^2} & \text{if } x_0 \leq \Delta\delta_{ij}(\mathbf{X}). \end{cases} \tag{3.22}$$

where the parameters, $n\epsilon_j, x_0, \beta_j$ and γ have been empirically adjusted. The potential has a flat bottom, with the width of $n\epsilon_j$. The flat bottom corresponds to the expected standard deviation of CamShift, to avoid overfitting in the simulation. The penalty function grows harmonically until a cut-off of x_0 and follows a somewhat flat hyperbolic tangent function after this. While there is no substantial theoretical backing

3.3 Sampling strategy for weight parameters

Since the nuisance parameters of the energy functions are unknown, they too must be sampled. The move used to update the value of the nuisance parameters must obey detailed balance:

$$p\left(w \to w'\right) = p\left(w' \to w\right) \tag{3.23}$$

The simplest Monte Carlo move is simply adding a number from a normal distribution with $\mu = 0$, this clearly obeys detailed balance, since the distribution is symmetric. For the weight parameters, γ and σ, of the Cauchy and Gaussian distributions, respectively, we found a variance of 0.05 in the normal distributed move to converge quickly and stably.

3.3.1 Molecular mechanics force field

One reasonable prior distribution for protein structure, $p(\mathbf{X})$, is the Boltzmann distribution, e.g.:

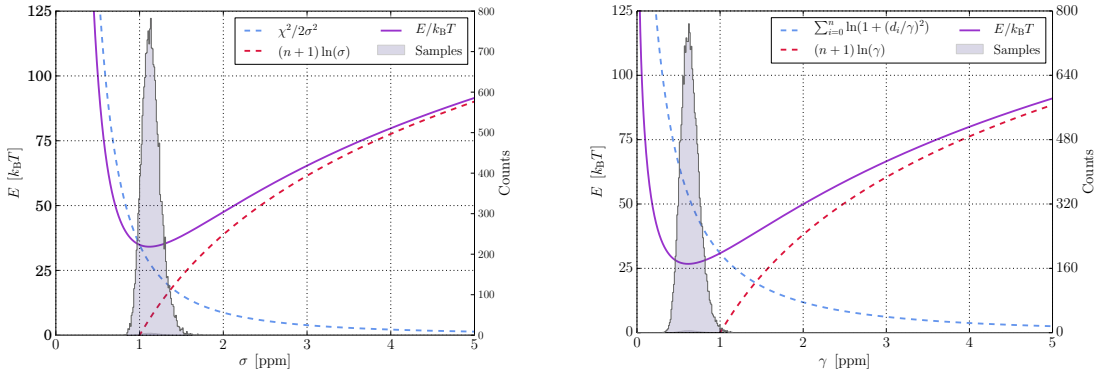
$$p(\mathbf{X}) \propto \exp\left(\frac{-E}{k_{\mathrm{B}}T}\right) \tag{3.24}$$

where E is the energy of the structure, $\mathbf{X}$ and k_{B} and T are Boltzmann's constant and the temperature, respectively. The energy of the structure is in this context usually approximated by a molecular mechanics force field that is taylor-made for protein simulations. PHAISTOS currently supports two different protein force field: The OPLS-AA/L force field with a GB/SA solvent term, and the coarse-grained PROFASI force field. The OPLS-AA/L is an all-atom force field with an additional solvation. The PROFASI force field is a coarse-grained force-field which assumes fixed bond-lengths and angles and furthermore has a very aggressive 4.5 Å cut-off of long-range interaction terms.

3.4 Results

3.4.1 Results – sampling of weight parameters

Figure 3.1 show a histogram of 100,000 sampled values of γ and σ for the NMR structure of Protein G (PDB-id: 2OED). No structural moves were used, and the results are thus temperature independent since the physical energy is constant. A total of 55 C^{α} experimental chemical shifts



(a) Gaussian distribution

(b) Cauchy distribution

Figure 3.1: Sampling of σ and γ for 2OED for Ca-chemical shifts. $n = 55$ and $\chi^2 = 69.7$.

were used in this example (RefDB-id: 2575), and CamShift was used to calculate the chemical shifts. The initial values of σ and γ was 10.0, in order to demonstrate the stable convergence using the simple move.

In both simulations, the sampling algorithm converges sampling around the minimum of the energy function. In both cases, these minima are in very good agreement with the values calculated by the test set that was used to validate the performance of CamShift. The largest sampled bins are centered on $\sigma = 1.26$ ppm and $\gamma = 0.63$ ppm for the Gaussian and Cauchy distributions, respectively. These number can be compared to the maximum likelihood estimates (MLE) obtained on the 7 protein benchmark set used to determine the accuracy of Camshift. Here the values are $\sigma = 1.3$ ppm and $\gamma = 0.7$ ppm for the Gaussian and Cauchy distributions, respectively.

3.4.2 Performance of energy functions

Here the performance of folding simulation using 11 different variations of the energy function derived and mentioned previously. All energy functions have been implemented in the CamShift module in PHAISTOS, which was also used to run all simulations. The test were carried out on Protein G and the engrailed homeodomain (ENHD). The reference structures were the structures 2OED and 1ENH. An overview of the different simulation types can be found in Table3.1. For each energy function, 20 independent simulations were carried out for a total of 50,000,000 MC steps each. Each simulation was initialized from a different random, extended strand. Maximum likelihood estimated (MLE) values of the σ and γ weight parameters estimated take from the 7-protein test set reported in reference [Ref XX]. For simulations where the weight parameter was sampled, an additional 500,000 Monte Carlo steps were carried out corresponding to the extra moves required to sample this weight (the computational overhead of these 500,000 moves is negligible). Chemical shifts were calculated using the CamShift module. All simulations used the PROFASI force field and sampling from either TorusDBN or TorusDBN-CS.

In two simulations, the bias was removed from the simulation, which corresponds to an unbiased simulation. Two reference simulations were carried out with no chemical shift energy-function, in order to analyze the effect of sampling from TorusDBN and the effect of the PROFASI force field. The simulations used a mix of 40% biased CRISP-moves, 10% biased pivot moves and 50% uniform side chain moves. The simulation was carried out in the multicanonical ensemble via Muninn. Minimum and maximum β-values were set to 0.3 and 1.05, and the temperature was set to 300K. In all simulations, the number of threads which had samples below thresholds of 5, 3, 2 and 1 Å CA-RMSD from the crystal structure was recorded. Similarly, the number of threads in which the lowest energy structure was below thresholds of 5, 3, 2 and 1 Å CA-RMSD

Table 3.1: Protocols used in the comparison of energy functions and success rates.

Energy type	Weight	TorusDBN-mode	Sampling Bias	Correct sampling[a]	Correct scoring[b]
Gauss	Fixed	Torus	Biased	20/20	2/6
Gauss	Sampled	Torus	Biased	0/0	0/0
Cauchy	Fixed	Torus	Biased	20/12	7/4
Cauchy	Sampled	Torus	Biased	20/5	6/1
Cauchy	Sampled	Torus-CS	Biased	20/20	4/2
Cauchy	Sampled	Torus	No bias	0/0	0/0
Cauchy	Sampled	Torus-CS	No bias	0/0	0/0
Square-well	Fixed	Torus	Biased	1/2	1/0
Marginalized	N/A	Torus	Biased	7/17	1/8
No CS	N/A	Torus	Biased	0/0	0/0
No CS	N/A	Torus-CS	Biased	8/10	2/0

[a] Number of threads with a CA-RMSD of < 5 Å (using all residues). Listed as xx for Protein G and yy for ENHD, i.e. xx/yy.
[b] Number of threads where the lowest energy sample has a CA-RMSD of < 3 Å (using all residues). Listed as xx for Protein G and yy for ENHD, i.e. xx/yy.

from the crystal structure was recorded. These figures are used to analyze whether sampling or correct energy scoring is are limiting factors in the particular simulations. The energy was calculated as the PROFASI energy multiplied by $k_{\mathrm{B}}T$ plus the chemical shift energy term plus the log-likelihood calculated from TorusDBN. An overview of these results can be seen in Fig. 3.2 (only simulations that had any samples below 5 Å CA-RMSD from the crystal structure are shown).

For both proteins, using a Gaussian model and sampling the σ uncertainty does not lead to meaningful values for σ. In short, PHAISTOS is able to generate a structure which has no difference between experimental and calculated chemical shifts for a certain atom type. Consequently, the value of σ converges to zero, which effectively freezes the structure in the simulation. The simulations in which the move-bias from TorusDBN and TorusDBN-CS was removed did not sample any structures below

For simulations using Gaussian or Cauchy types of energy function all thread had samples below 5 Å CA-RMSD from the crystal structure for Protein G and between 5-20 for ENHD The simulation with the square-well potential only 1 thread had samples below 5 Å for Protein G and only 2 for ENHD. For the simulation with marginalized weight parameters, the same figures were 7 and 17, respectively. The reference simulations with no chemical shift in the energy function had no samples below 5 Å for biased sampling from TorusDBN, but 8 and 10 threads below 5 Å for biased sampling from TorusDBN-CS.

Comparing the number of threads for which the lowest energy sample was below 3 Å CA-RMSD from the crystal structure. For both proteins, using fixed weights is somewhat better than using sampled weights with the Cauchy distribution. The result for the square-well potential cannot be interpreted to a statistical significance because only one and two threads were close to the correct fold, but one thread correctly identified the folded state below 3 Å CA-RMSD as the lowest energy for Protein G.

In conclusion, the Gauss and Cauchy perform well in sampling and scoring. The fixed MLE weights seem to be work equally well to sampling weights for the cauchy distribution, with no substantial differences. The performance of the energy with marginalized weights performed worse in guiding the sampling, but well in scoring samples for ENHD. The square-well potential did not improve the sampling much. The reason why it has previously been shown to work well,

might be that it was combined with a better force-field (AMBER03) to which it was specifically designed. One clear conclusion is that it is useful to not remove the bias from TorusDBN, and keeping the TorusDBN-CS bias seems guide folding significantly more, even though this formally constitutes is double-counting of effect of knowledge about chemical shifts.

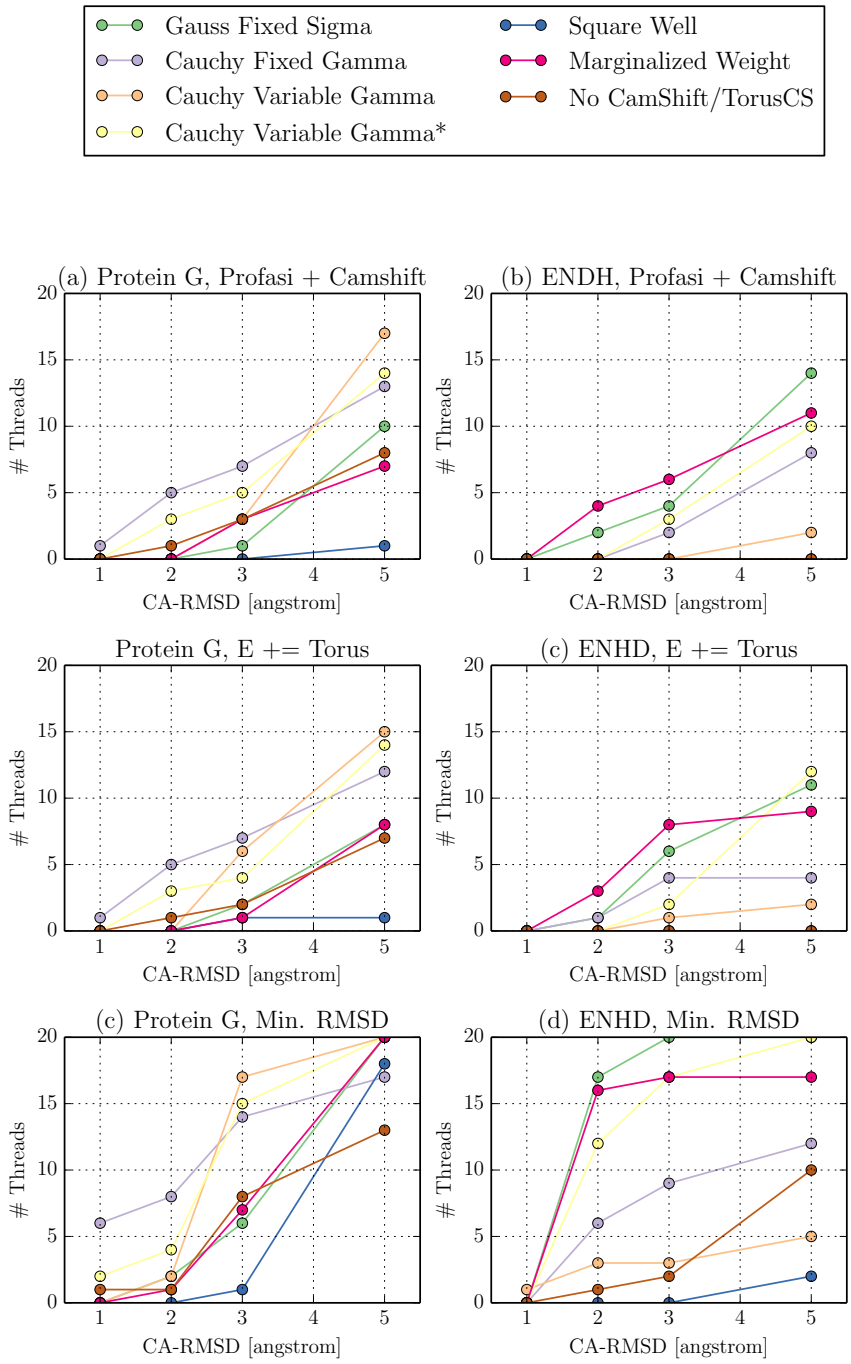


Figure 3.2: Overview of folding simulations using 7 different chemical shift energy types. Sampling was biased by TorusDBN and the PROFASI energy term was used as well. In (a) and (b) the number of threads where the lowest energy samples are under thresholds of 1, 2, 3 and 5 Å CA-RMSD from the crystal structure is plotted. The energy here is calculated as the PROFASI energy multiplied by k_BT plus the chemical shift energy term. In (c) and (d), the log-likelihood from TorusDBN has been added to the total energy. In (e) and (f), the number of threads in which samples are found below under thresholds of 1, 2, 3 and 5 Å CA-RMSD from the crystal structure is plotted. *In this simulation TorusDBN-CS is used instead of TorusDBN.

Chapter 4

Graphical User Interface for PHAISTOS

Setting up simulations in PHAISTOS requires expert knowledge about the program. Firstly, while all modules and settings have reasonable default settings, there are still many things that cannot be specified via default alone, and secondly, the complete list of settings in PHAISTOS is around 2500 options that must be set or taken as default values.

In order to make PHAISTOS more available to new users, I wrote a GUI can set up most simulations for most of the simulations covered by this thesis. The GUI for PHAISTOS is aptly named Guistos and is written in Python 2.x using TkInter.

Using the GUI the user is only presented with the three most basic choices for setting up the simulation. These are (1) choice of energy terms, (2) type of Monte Carlo simulation and finally (3) a selection of Monte Carlo moves. Setting up these via Guistos is discussed next.

Energy Options

Firstly, the Energy Options section allows the user to select the molecular mechanics force field. Currently two force fields are supported in PHAISTOS, which are the OPLS-AA/L force field with a GB/SA solvent model, and the PROFASI coarse grained force field. Use of the PROFASI force field requires the Monte Carlo moves to restraint the bond angle and lengths in the protein to Engh-Huber standard values. This is automatically done if the PROFASI force field is selected. Conversely, the OPLS-AA/L force field includes energy terms for bond angles and lengths and these are degrees of freedom in the simulation if the OPLS-AA/L force field is selected.

Additionally, the Energy Options section allows the user to add restrains from one type spectroscopic data. Currently energy terms based on CamShift 1.35 and ProCS are supported. These options requires a NMR-STAR formatted file containing experimental chemical shifts.

Monte Carlo Options

This section allows the user to select the four types of Monte Carlo simulation offered by PHAISTOS and the only the most basic options to set up that particular simulation: Metropolis-Hastings offers the choice of a constant temperature (in Kelvin). Muninn and Simulated Annealing offer the choice of a temperature range (in Kelvin), and additionally Muninn offers the choice between multicanonical or $1/k$ sampling. Greedy Optimization does not offer any customizable option.

Monte Carlo Move Sets

Selecting a good mix of the different Monte Carlo moves offered by PHAISTOS can significantly speed up convergence of a simulation, compared to using an inferior move set. Choosing a good

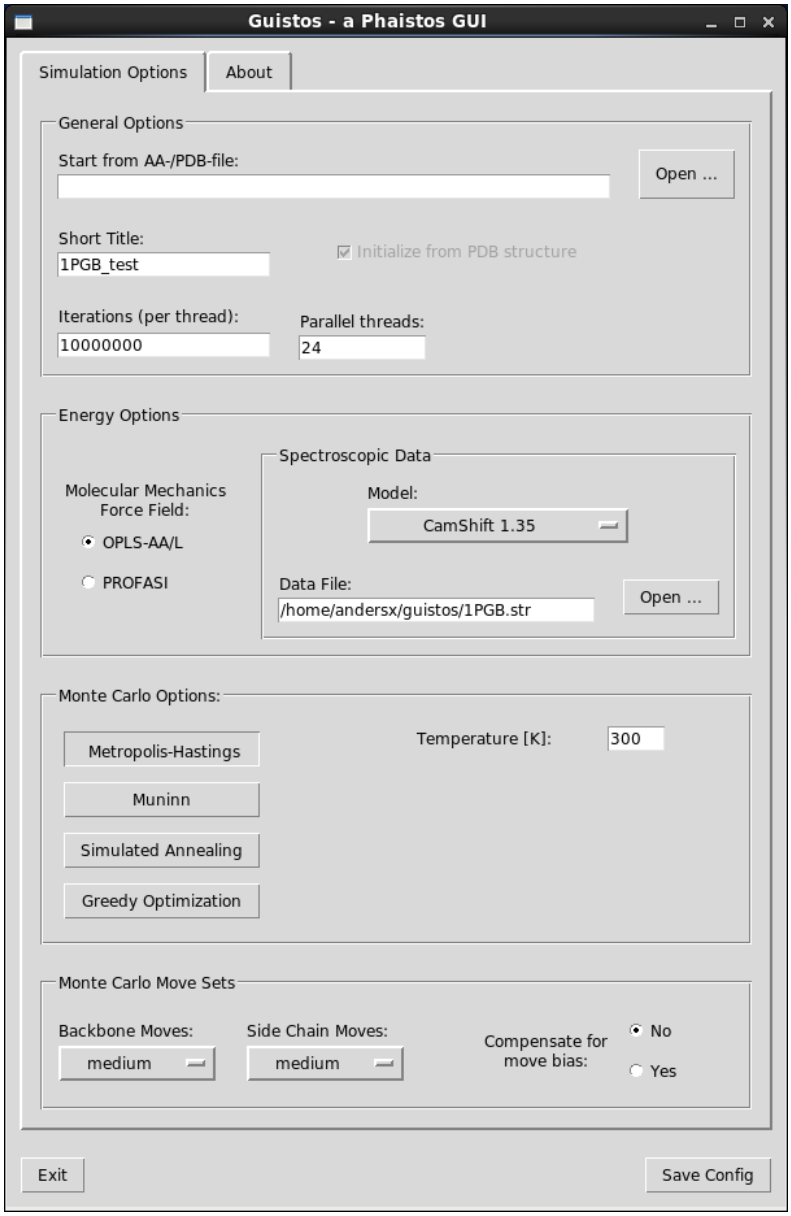


Figure 4.1: Screenshot of Guistos

set of moves is in the opinion of this author currently somewhere in between black art and sheer luck, and requires a good deal of experience with simulations in PHAISTOS.

To make it easier for new users, three move sets have been predefined using the experience of this author. These are named "small", "medium" and "large". The "small" move set is intended for uses such as refinement or sampling around a compact native state, while the "medium" move set is intended for folding simulations that start from extended, but are expected to also sample a native state, and finally the "large" move set is intended for sampling conformational space quickly, but will have problems with sampling compact structures. All move sets sample from TorusDBN (backbone angles) and BASILISK (side chain angles), and an option to remove this bias is also present.

Using Guistos

Guistos is freely released under the open source two-clause BSD-license, and can be downloaded from `https://github.com/andersx/guistos`. A screenshot of Guistos can be seen in Fig. 4.1. After specifying all relevant settings in the Guistos window, a configuration-file is saved by pressing the "Save Config" button. A simulation in PHAISTOS can the be executed via the following command:

```
1 ./phaistos --config-file my_simulation.config
```

Chapter 5

Structure Based Prediction of Protein Chemical Shifts

Chapter 6

Determined protein structures

This section describes all test-targets which I have attempted to fold using the methodologies presented in the previous chapters.

6.1 Barley Chymotrypsin Inhibitor II

An especially interesting target in this study is the barley chymotrypsin inhibitor II (CI-2). CI-2 is a 63 residue protein which consists of an α-helix which connects via a very flexible handle to a small β-sheet region.

The chemical shifts data was obtained using a fully automated procedure. The ADAPT-NMR [REF XX] protocol was used to record all necessary NMR data and automatically assign the chemical shifts. Data collection and assignment was completed in only 11 hours with minimal human intervention. As we demonstrate, a structure could be determined computationally from these chemical shifts in only two days running on 12 cores.

6.1.1 Computational methodology

Several folding protocols were tried for this protein. All runs were performed as 72 independent trajectories which ran for 50 mio MC steps (iterations). Sampling was carried out using either TorusDBN or TorusDBN-CS to bias the backbone moves and the PROFASI force field was used in all simulations. Three simulations used an energy function based on CamShift using a cauchy distribution with variable γ value as energy function. Additionally, three simulations used a potential on the radius of gyration to restrict the sampling to only compact structures. MUNINN was was set to multicanonical sampling and the thermodynamic beta-range was set to between 0.6 and 1.1, corresponding to a temperature range of 272K to 500K. The MC moves were set to 49% CRISP moves, 2% pivot moves and 49% uniform side chain moves.

6.1.2 Folding results

Three of the 7 attempted simulation types sample structures close to the experimental X-ray structure 1YPA (here loosely defined as a CA-RMSD $<$ 5 Å for all CA atoms. Results are summarized n thable None of the simulations that sample from TorusDBN (not chemical shift biased) are able to sample the correct fold.

Furthermore, it was noted, that simulations that sample from either TorusDBN or TorusDBN-CS with only the PROFASI force field as energy function do not generate compact structures. To overcome this deficiency, additional simulations were carried out using a radius of gyration potential. In the case of sampling from TorusDBN-CS, the radius of gyration potential is enough to get a few samples with the correct fold. Here four of 72 threads would generate the correct fold,

Table 6.1: Protocols used in the folding of the CI-2 protein and success rates.

Sampling	Force Field	CS Energy	Correct fold[a]	Iterations/day[b]
TORUS-CS + PP[c]	PROFASI	CamShift	13	10×10^6
TORUS-CS	PROFASI	CamShift	15	11×10^6
TORUS	PROFASI	CamShift	0	11×10^6
TORUS-CS + PP[c]	PROFASI	None	4[d]	49×10^6
TORUS-CS[e] + PP[c]	PROFASI	None	0	49×10^6
TORUS-CS	PROFASI	None	0	49×10^6
TORUS	PROFASI	None	0	49×10^6

[a] Number of threads with a CA-RMSD of < 5 Å (using all residues).
[b] Numbers are *per* thread.
[c] PP denote the use of a radius of gyration potential.
[d] Structures with the lowest energy did not correspond to the native structure in this run.
[e] This run was carried out using TorusDBN-CS trained using only high-quality X-ray structures.

but unfortunately the lowest energy structures were found around 8-11 Å CA-RMSD. Evidently, the PROFASI force field alone is not accurate enough to describe the native CI-2 structure. Three simulations were performed with an energy term based on CamShift in addition the PROFASI force field. Demonstrably, the increased accuracy from a better energy function cause increased sampling around the native state.

Due to a very flexible region of CI-2 (residues 33 to 42), and somewhat flexible tails the residue range used to calculate CA-RMSD values is restricted to residue 4-34,43-63 in the following. All runs were carried out on 3 24-core AMD Opteron 6172 servers running at 2.1 GHz.

A run similar to the most successful was also run carried out on a faster a 12-core Intel X5675 node running at 3.07 GHz (using new random seeds). This simulation took two days, with a total of 2 out of 12 threads successfully identifying the native structure as having the lowest energy. This simulation yielded a lowest energy structure a 2.76 Å CA-RMSD from the X-ray structure, and a lowest RMSD structure at 1.11 Å.

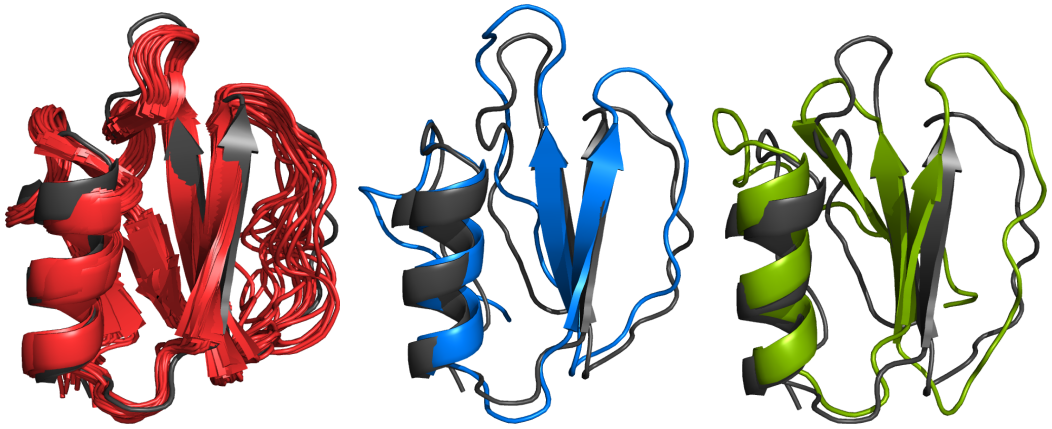
The lowest RMSD was further refined by Lars Bratholm to a CA-RMSD of only 1.1 which took 24 hours on 8 cores.

6.2 Folding of small proteins (<100 AA)

Table 6.2: Folded structure.

Name	Lengh	Type	PDB	RefDB	RMSD-range	Final RMSD
Protein G	56	a/b	2OED	2575	All	1.0
Engrailed Homeodomain	61	B	1ENH	15536	8-53	1.1
CI-2	63	a/b	1YPA	N/A[a]	4-34,43-63	2.6
FF Domain	71	a/b	1UZC	5537	11-67	10.2*
Ubiquitin	76	a/b	1UBI	17769	1-70	3.8

[a] Using automatically assigned data obtained from Kaare Theilum (personal communitcation - see `https://github.com/andersx/cs-proteins`).



(a) NMR structure (red) (b) Lowest RMSD structure (blue) (c) Lowest energy structure (green)

Figure 6.1: Structures compared to the X-ray structure 1YPA. All structures are aligned using the residues 12-32,43-52. (a) shows the 3CI2 structure NNR structure. Note the flexible domain which is excluded from the fit-range. (b) Shows the lowest RMSD structure (1.113 ÅRMSD). (c) shows the lowest energy sample (2.76 ÅRMSD).

6.3 Folding of larger proteins (>100 AA)

This section presents folding results on a set of larger proteins (>100 AA) with known structures. It is worth to note, that using sparse NMR data, only three structures >200 residues have been determined: Alg13 (201 AA), Rhodopsin (225 AA) and MBP (376 AA) using the Rosetta program with the "resolution-adapted structural recombination" (RASREC) protocol

Alg13 was solved using backbone chemical shifts only 52 NOE, to an CA-RMSD of 4 Å to the experimental NMR structure (2jzc). Rhodopsin was folded to an CA-RMSD of 1.6 Å to the X-ray structure using 215 NOE restraints, backbone chemical shifts chemical shifts and RDCs. The MBP protein is a two-domain protein of 376 residues. MBP was folded to an RMSD of 3.6 Å using 1235 NOE restraints, backbone chemical shifts chemical shifts and RDCs. The NOEs corresponded to 55% yield of restraints, which, however mostly were not automatically assigned. An attempt to use only automatically assigned NOEs yielded 455 restraints, which corresponds to a yield of 20%. Using these, however, the MBP structure could only be determined to a total CA-RMSD of 12.3 Å. The N-terminal domain was converged to 2.7 Å, but the C-terminal domain and the angle between the two domains was incorrectly folded.

6.3.1 Folding protocol

The folding protocols used in the following was a two stage simulation. First folding, then refinement.

Initial folding stage:

```
./phaistos --aa-file rhodopsin.aa \
  --iterations 50000000 \
  --threads 72 \
  --monte-carlo-muninn 1 \
  --monte-carlo-muninn-min-beta 0.6 \
  --monte-carlo-muninn-max-beta 1.1 \
  --monte-carlo-muninn-independent-threads 1 \
```

```
  --monte-carlo-muninn-weight-scheme multicanonical \
  --backbone-dbn-torus-cs 1 \
  --backbone-dbn-torus-cs-initial-nmr-star-filename \
                                     rhodopsin.str \
  --energy-profasi-cached 1 \
  --energy-isd-dist 1 \
  --energy-isd-dist-likelihood square_well \
  --energy-isd-dist-data-filename noe_ilv.txt \
  --energy-isd-dist-sample-gamme 0 \
  --energy-isd-dist-sample-sigma 0 \
  --energy-isd-dist-weight 0.0078125 \
  --move-backbone-dbn 1 \
  --move-backbone-dbn-weight 0.08 \
  --move-backbone-dbn-implicit-energy 1 \
  --move-crisp-dbn-eh 1 \
  --move-crisp-dbn-eh-weight 0.42 \
  --move-sidechain-uniform 1 \
  --move-sidechain-uniform-weight 0.5
```

Refinement protocol:

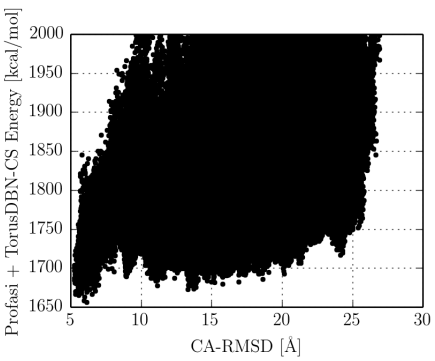
The

```
./phaistos --pdb-file rhodopsin_lowest_energy1.pdb \
  --init-from-pdb 1 \
  --iterations 2000000 \
  --threads 8 \
  --monte-carlo-metropolis-hastings 1 \
  --monte-carlo-metropolis-hastings-declash-on-reinitialize 0 \
  --backbone-dbn-torus-cs 1 \
  --backbone-dbn-torus-cs-initial-nmr-star-filename \
                                     rhodopsin.str \
  --energy-profasi-cached 1 \
  --energy-pp-compactness 1 \
  --move-backbone-dbn 1 \
  --move-backbone-dbn-weight 0.08 \
  --move-backbone-dbn-implicit-energy 1 \
  --move-crisp-dbn-eh 1 \
  --move-crisp-dbn-eh-weight 0.42 \
  --move-sidechain-uniform 1 \
  --move-sidechain-uniform-weight 0.5
```

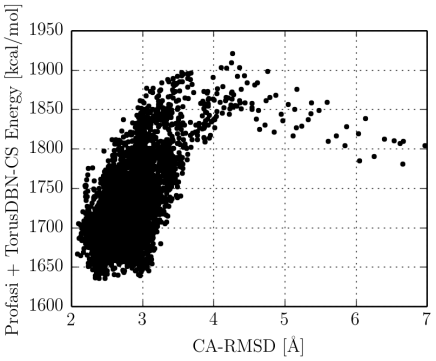
6.3.2 Rhodopsin (225 residues)

Since the ILV(W) data set used by Rosetta was not available, simulated ILV restraints from the PDB structure 1h68 was used instead. A very conservative simulation of only 63 synthetic NOE restraints was used, which corresponds to around 4% assigned long-range contacts. The initial folding stage converged to around 6.8 as the lowest energy cluster - see Fig. 6.2). Two threads out of 72 converged to this native-like cluster.

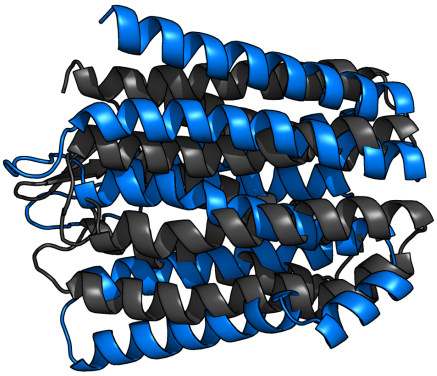
The refinement stage with only 63 NOEs, however, did not converge to a lower CA-RMSD. The Rosetta data set includes 213 contacts, and a similar, less conservative synthetic NOE data was simulated from the PDB structure 1h68, in order to see, if an increased number of NOE restraints could drive the RMSD down. This resulted in 195 NOE restraints, which corresponds to 13%



(a) Energy-scoring during folding stage.



(b) Energy-scoring during refinement stage.



(c) Folding stage lowest energy sample (blue). 7.8 Å CA-RMSD.

(d) Refinement stage lowest energy sample (blue). 2.5 Å CA-RMSD.

Figure 6.2: Some caption

Table 6.3: Folded structure.

Name	Lengh	Type	PDB	BMRB	RMSD-range	RMSD
APO-LFABP	129	a/b	1LFO	15429[a]	All	
Prolactin	199	B	1RWS	5599	6-183	
Top7	120	a/b	2MBL	19404	5-104	
MSRB	151	a/b	3E0O	17008	36-105	
WR73	183	a/b	2LOY	16833	1-36,66-181	
HR4660B	174	a/b	2LMD	1870	16-162	
Rhodopsin	219	B	2KSY	16678	All	3.2

[a] Data was obtained from RefDB.

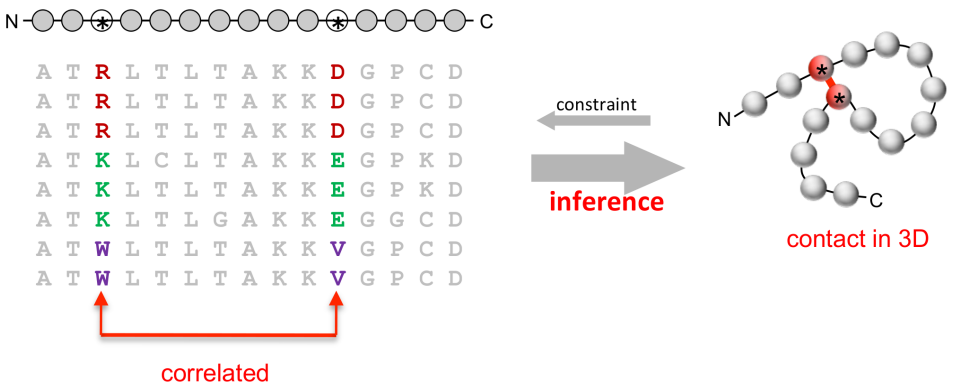


Figure 6.3: Evolutionary constraints

assigned NOEs. Using this, larger, set of restraints during the refinements gave a substantial decrease in CA-RMSD for the lowest energy sample to 3.6 Å.

It must be noted, however, that in both the folding and refinment stages, samples are obtained with less than 2650 kcal/mol (calculated as the Profasi force field energy plus the likelihood from Torus-DBN-CS given the experimental chemical shifts), while the native energy is 2730 kcal/mol. This discrepancy is likely due the lack of accurate non-local information in the Profasi and Torus-DBN-CS models.

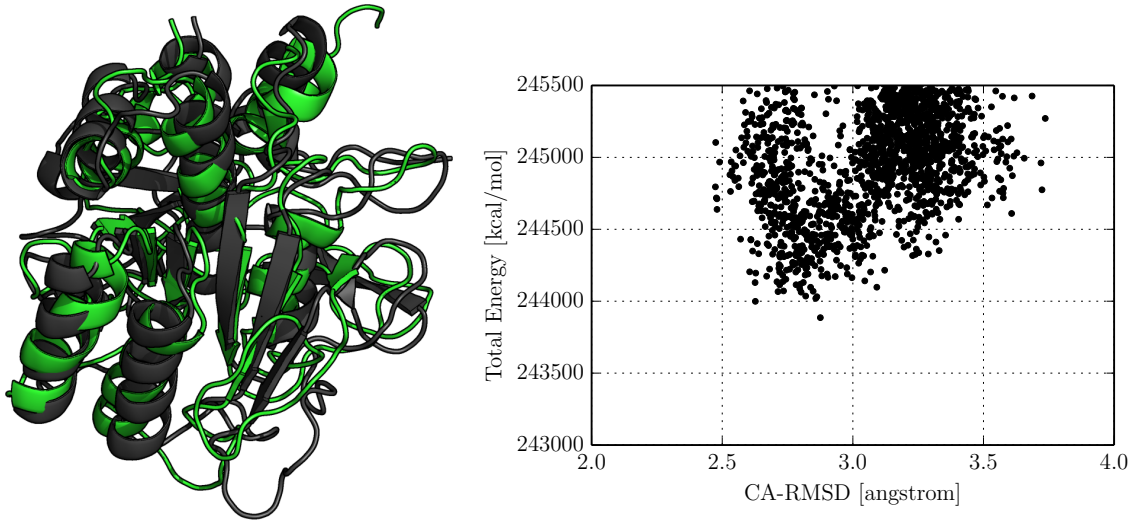
Further refinement was attempted with the CamShift energy-term instead of the Torus-DBN-CS model, but was found to be too slow to be practically feasible.

6.4 Evolutionary distance constraints

As discussed previously, it is increasingly difficult to obtain sufficient distance restraints as the size of the protein increases. A recently developed methodology uses sequence analysis to infer residue contacts in 3D space.

In brief, the method works by identifying sequence co-variation, which retains favorable contacts between residues. This way, pair of residues which are probable to be close in 3D space can be identified. The procedure is briefly summarized in Fig. 6.3.

In this proof-of-concept study, 270 contacts were obtained a multiple-sequence alignment using the EVfold program (Wouter Boomsma, personal communications) for the 269 residue protein Savinase. The restraints were simply treated as NOE restraints using existing code. A similar simulation to that which folded Rhodopsin was adopted. In terms of computational resources, these were increased to 100 threads and 75×10^6 iterations, compared to only 72 threads and



(a) Lowest RMSD sample

(b) Lowest energy sample

Figure 6.4: Refinement stage of the savinase simulation. The lowest energy sample has a CA-RMSD of 2.9 Å.

50×10^6 iterations for the Rhodopsin simulation. One thread identified a native-like structure. The folding simulation yielded a lowest energy

Bibliography

[Cavalli et al., 2007] Cavalli, A., Salvatella, X., Dobson, C. M., and Vendruscolo, M. (2007). Protein structure determination from nmr chemical shifts. *Proc. Natl. Acad. Sci.*, 104:9615–9620.

[Jack and Levitt, 1978] Jack, A. and Levitt, M. (1978). Refinement of large structures by simultaneous minimization of energy and r factor. *Acta. Cryst.*, A34:931–935.

[Rieping et al., 2005] Rieping, W., Habeck, M., and Nilges, M. (2005). Inferential structure determination. *Science*, 308:303–306.

[Robustelli et al., 2009] Robustelli, P., Cavalli, A., Dobsom, C. M., Vendruscolo, M., and Salvatella, X. (2009). Folding of small proteins by monte carlo simulations with chemical shift restraints without the use of molecular fragment replacement or structural homology. *J. Phys. Chem. B*, 113:7890–7896.

[Robustelli et al., 2010] Robustelli, P., Kohlhoff, K., Cavalli, A., and Vendruscolo, M. (2010). Using nmr chemical shifts as structural restraints in molecular dynamics simulations of proteins. *Structure*, 18:923–933.

[Shen et al., 2008] Shen, Y., Lange, O., Delaglio, F., Aramini, J. M., Liu, G., Eletsky, A., Wu, Y., Singarapu, K. K., Lemak, A., Ignatchenk, A., Arrowsmith, C. H., Szyperski, T., Montelione, G. T., Baker, D., and Bax, A. (2008). Consistent blind protein structure generation from nmr chemical shift data. *Proc. Natl. Acad. Sci.*, 105:468–4690.