Abstract

In this thesis an algorithm for simulating the equilibrium behavior of a large number of protein sequences at fixed temperature in a single run is investigated. The method, which works by allowing Monte Carlo updates of the protein sequence, is implemented and tested for a simple reduced-representation continuous protein model with three different types of amino acids.

The effectiveness of the model is related to that of a naive fixed-sequence Monte Carlo algorithm, by comparing thermalization times and statistical errors. The new algorithm is found to have considerably shorter thermalization times, and statistical errors of similar or lower magnitude for most sequences investigated.

Finally, the method is used to investigate the sequence-structure relationship for short model proteins with eight amino acids. Sequences folding to certain structures were mapped, and the number of sequences folding to each were determined, as well as the number of mutational transitions between them.
1 Introduction

Proteins are one of the most common types of macro-molecules in living organisms. They play an important role in several biological and biochemical processes, e.g. signal transmission and transport, as enzymes, or as part of the immune system. In addition, proteins provide structure for biological materials such as hair, nails or cartilage. [1]

A protein can be described by its amino acid sequence. When biologically active, however, the three-dimensional structure of a protein is very important. One of the main questions concerning proteins and their biological role is therefore the connection between sequence and structure.

One way to investigate the properties of various proteins is to perform simulations of simplified model proteins. It is then of importance to be able to effectively sample the properties of these model sequences.

The aim of this work is to implement and test a new method for simultaneous sampling of the equilibrium ensembles of a large number of different sequences in a simplified protein model. In order to demonstrate the usefulness of this method we then use it to investigate the connection between sequence and structure for short, eight amino acids long model proteins.

2 Background

2.1 Proteins and Amino Acids

The basic building blocks of proteins are amino acids. Each amino acid consists of an amine group (-NH₂) and a carboxylic acid group (-COOH) which each bind to a carbon atom. Also binding to this carbon atom is a hydrogen atom and a a side chain. This side chain determines the type of the amino acid. A graphical representation of an amino acid can be seen in figure 1.

![Amino Acid](image)

Figure 1: The chemical structure of an amino acid. R denotes the side chain.

The protein is a linear chain of amino acids, held together by covalent bonds between the carboxylic acid group of one amino acid and the amine group of the next. When forming these bonds, a water molecule is cleaved off. In this way, the protein forms a backbone which, up to differences in length, is the same for all proteins. The shorter side chains, on the other hand, are unique for each protein. Figure 2 shows a graphical representation of the protein chain.

At the most basic level, a protein can be described through its sequence of amino acids, often called the primary structure.
The diversity of protein structures and functions observed in biological systems arise from the different physical properties of the different amino acids. Human proteins contain 20 different kinds of amino acids. However, roughly speaking, most amino acids can be classified in one of two groups, hydrophobic or polar, depending on whether or not their side chains are able to form hydrogen bonds with the surrounding water [1, pp. 1-7, 157].

Two amino acids with special geometric properties are glycine and proline. For this thesis, the most important is glycine. Glycine is unique in that it does not have a side chain, but only a hydrogen atom. This gives glycine a greater flexibility, and it is often found in “turns,” where the protein changes its secondary structure.

2.2 Protein Folding

The function of a protein is closely related to its three-dimensional structure. Most biologically active proteins fold into a single, well-defined structure, which is called the native state of the protein. The native state is uniquely determined by the sequence of a protein. However, the relation is far from trivial since there are several effects which play important roles in protein folding [2] [1, pp. 139-148, 157-162]. We will describe the most important of these below.

One of the major driving forces behind protein folding is the so called hydrophobic effect. Since hydrophobic amino acids cannot form hydrogen bonds with the surrounding water, they break the web of hydrogen bonds surrounding them, effectively giving rise to an attractive force between hydrophobic amino acids. Because of this proteins typically form a core of hydrophobic side chains, shielded from water by the polar amino acids.

Another important stabilizing contribution is hydrogen bonding between the amine and carboxylic acid groups of different amino acids. These hydrogen bonds give a major contribution to the stability of the local, so called secondary, structure of the protein.

In addition to the contributions discussed above, electrostatic interactions and van der Waals bonds also contribute to the folding behavior, however these interactions are typically not as important for the folding process.

The spatial structure of a protein is usually separated in what is called secondary and tertiary structure. The secondary structure describes the local conformation of the protein, whereas the tertiary describes the complete three-dimensional
fold. The proteins studied herein are too short to have a tertiary structure, and we will therefore concentrate on the secondary structure below [1, pp. 171-172, 182-187].

One of the most common types of secondary structure is the $\alpha$-helix. As the name implies the protein backbone forms a helical shape with the side chains on the outside. Typically, the protein makes one turn every 3.5 amino acids. The $\alpha$-helix is stabilized by hydrogen bonds between the amine group of an amino acid and the carboxylic group of the amino acid four positions earlier in the chain.

Another common secondary structure is the $\beta$-structure. These are characterized by two or more parallel protein strands connected by hydrogen bonds. Typically, double hydrogen bonds exist between every second pair of amino acids. The proteins we will study are so short that only a single turn of the chain will be possible. This very short piece of $\beta$-like structure is often referred to as a $\beta$-hairpin.

2.2.1 Thermodynamic stability

Later in the thesis, we will be interested in quantifying the stability of the native protein fold. To do this, we may consider the protein as a two-state system, with a folded and an unfolded state. Each state is characterized by its free energy, $F$, and the probability of being in a specific state is proportional to the Boltzmann exponential $e^{-\beta F}$, with $\beta = 1/k_BT$ and $T$ denoting the temperature. One measure of the thermodynamic stability is the free energy difference $\Delta F = F_{\text{folded}} - F_{\text{unfolded}}$ [3, pp. 223-226]. From the Boltzmann exponential we easily get that

$$\frac{P_{\text{folded}}}{P_{\text{unfolded}}} = e^{-\beta \Delta F} \quad (1)$$

Inverting this expression, and using $P_{\text{unfolded}} = 1 - P_{\text{folded}}$, the free energy difference can be expressed as

$$\beta \Delta F = -\log \frac{P_{\text{folded}}}{1 - P_{\text{folded}}} \quad (2)$$

2.3 Protein Evolution

A central question in protein research is how proteins evolve. New proteins can arise through mutations of already existing proteins. There are several different types of mutations that can occur. Simple point mutations of single amino acids, for example, are thought to be an important evolutionary driving force. Therefore, the relationship between sequence and structure is important for understanding the mutational evolutionary process.

While the native state of a protein is uniquely determined by its sequence, the reverse is not necessarily true. Often, proteins with quite different sequences can fold to the same structure. The set of all sequences folding to the same
native structure is usually called the *neutral set* of the structure. The largest connected part of this set is called the *neutral net* \([4]\).

The sequence which can tolerate the largest number of point mutations in a neutral net is called the *prototype sequence* of the neutral net.

Despite the high resistance to random mutations, proteins can sometimes be very susceptible to specific mutations. Recent experiments have shown that sometimes a single mutation can cause the protein to completely change its conformational structure. This phenomenon is usually referred to as fold switching. It is unclear how common fold switching is, but clearly it can play an important role in protein evolution.

In discussing the relationship between different sequences, especially for simplified models, it is useful to introduce the Hamming distance, \(\Delta H\), which is defined as the number of positions in which the two sequences differ.

We can also quantify the mutational stability of a sequence as the number of point mutations on the sequence which causes it to remain in the same neutral net.

Studies of lattice proteins have shown a correlation between mutational stability and thermodynamic stability, see e.g. \([5]\). Note that this study used somewhat different measures of the two kinds of stability.

The concepts introduced in this section will be of importance towards the end of this thesis, when we discuss the sequence-structure relation in a simple protein model.
3 Model and Methods

3.1 The Model

In this thesis, we use an off-lattice protein model developed in reference [6]. In order to reduce the computational complexity of the simulations, the model used does not take into account every feature of a real protein.

The model proteins are geometrically quite reminiscent of real proteins, but with each amino acid consisting of only seven atoms. The backbone is represented in full detail, whereas the side chain consists of a single atom. In order to make it more bulky, the side chain atom was given a larger atom radius.

Along with the reduced side chain, the number of types of amino acids were also lowered. The model contains three different amino acids. One type of amino acid (h) has a hydrophobic side chain, whereas another (p) has a polar one. In addition, Glycine (G) was included as a separate amino acid.

As most bond lengths and angles are quite well defined in nature, they are held constant in the model. In each amino acid, only the torsional angles along the backbone are allowed to vary. This gives two degrees of freedom per amino acid in the protein.

The energy of the model protein can be written as a sum of four terms, \( E = E_{hp} + E_{hbond} + E_{exvol} + E_{local} \), each representing a specific type of interaction.

The hydrophobic energy is represented as an attractive force between pairs of hydrophobic side-chain atoms. The potential is (negative) Gaussian, with a natural distance of \( \sigma_{hp} = 5 \text{Å} \):

\[
E_{hp} = -k_{hp} \sum_{ij} e^{-\left(r_{ij} - \sigma_{hp}\right)^2 / 2}
\]  

where \( r_{ij} \) is the separation between the atoms.

Pairs with a separation of two or fewer amino acids are not included in the sum.

The second term is included to simulate hydrogen bonds, and is dependent on the angles between amine and carboxylic acid groups in addition to the distance between them:

\[
E_{hbond} = k_{hbond} \sum_{ij} \gamma_{ij} \left[ 5 \left( \frac{\sigma_{hb}}{r_{ij}} \right)^{12} - 6 \left( \frac{\sigma_{hb}}{r_{ij}} \right)^{10} \right] (\cos \alpha_{ij} \cos \beta_{ij})^{1/2}
\]  

where \( \alpha_{ij} \) is the N-H-O angle while \( \beta_{ij} \) is the H-O-C angle. The terms are included only if both \( \alpha_{ij} \) and \( \beta_{ij} \) are greater than 90°, and the two ends are separated by at least two amine groups. \( \gamma_{ij} \) depends on the amino acids involved, and \( \sigma_{hb} \) is a parameter. \( \gamma_{ij} \) is a parameter used to dis-encourage hydrogen bonding with glycine, by making these terms less energetically favorable. This is done to increase the chances of glycine acting as a breaker of secondary structure.

The third term is a so-called excluded-volume term. This term is included to discourage overlap between atoms. It has the form
\[
E_{\text{exvol}} = k_{\text{exvol}} \sum_{i<j} \left( \frac{\lambda_{ij}(\sigma_i + \sigma_j)}{r_{ij}} \right)^{12}
\]

where \( \lambda_{ij} \) is an atom-dependent scale factor, and \( \sigma_i \) is the radius of each atom. The sum is to be taken over all pairs of atoms.

Finally, there is a Coulomb interaction term between backbone atoms from the same amino acid. It represents the local energy arising from the charge distribution on the backbone:

\[
E_{\text{local}} = k_{\text{local}} \sum_{I} \sum_{i<j} \frac{q_{i}q_{j}}{r_{ij}}
\]

where the \( q_i \) are the partial charges, and \( r_{ij} \) is the distance between atoms \( i \) and \( j \). The outer sum is taken over all amino acids \( I \), while the inner one is taken over all atoms in that amino acid.

3.2 The Metropolis Algorithm

To sample the thermodynamic properties of a protein, we use the Metropolis algorithm which is a type of Monte Carlo (MC) method \[7\]. Using the Metropolis algorithm we sample different probability distributions taken from statistical mechanics.

The Metropolis algorithm can be thought of as a random walk in the space from which we want to sample, e.g. the space of all conformations of our model proteins. The algorithm can then be described by a set of transition probabilities, \( W(r \rightarrow r') \), which specify the probability of visiting state \( r' \) at step \( n+1 \), given that state \( r \) was visited at step \( n \). The Metropolis algorithm is a Markov process, i.e. the transition probabilities depend only on the current state.

The problem of sampling the correct distribution can now be solved by specifying a suitable set of transition probabilities. It can be shown that as long as the following two conditions are fulfilled, the sampled distribution will always approach the desired distribution \( P(r) \) \[8, pp. 142-146\].

1. The transition probabilities have to be ergodic: starting from any position in structure space it should be possible to reach any other position.
2. The limit distribution has to be stationary: \( P^{(n+1)}(r) = P^{(n)}(r) \) if \( P^{(n)} \) is the limit distribution.

The major difficulty typically lies in achieving the second condition. In the Metropolis algorithm, this is achieved by demanding that so called detailed balance is fulfilled. Mathematically, this condition can be written \[9, pp. 824-827\]

\[
P(r)W(r \rightarrow r') = P(r')W(r' \rightarrow r)
\]

If the system is in the equilibrium at time \( n \), the distribution at time \( n + 1 \) becomes
\[
\sum_{r'} W(r' \rightarrow r) P(r') = \sum_{r'} \frac{P(r)}{P(r')} W(r \rightarrow r') P(r') = P(r) \sum_{r'} W(r \rightarrow r') = P(r)
\]
where the last equality follows since the system has to be in some state after the update.

In order to fulfill detailed balance the transition probability is now written in the form

\[
W(r \rightarrow r') = F(r \rightarrow r') A(r \rightarrow r').
\]  
(8)

The first factor is called the proposal probability, and can be chosen arbitrarily (as long as it is ergodic). The in this way proposed update will then be accepted with a probability \(A\), chosen so that detailed balance is fulfilled. If we take the acceptance probability for transitions to states with higher limit probability to unity, the detailed balance condition (equation 7) becomes (with \(P(r) > P(r')\))

\[
P(r) F(r \rightarrow r') A(r \rightarrow r') = P(r') F(r' \rightarrow r).
\]  
(9)

This gives the acceptance probability for transitions to states with higher energy. The two cases can be summarized by the expression

\[
A(r \rightarrow r') = \min \left( 1, \frac{P(r') F(r' \rightarrow r)}{P(r) F(r \rightarrow r')} \right).
\]  
(10)

Finally, we note that if the proposal probabilities are chosen symmetrically, i.e. \(W(r \rightarrow r') = W(r \rightarrow r)\), the form of 10 becomes particularly simple:

\[
A(r \rightarrow r') = \min \left( 1, \frac{P(r')}{P(r)} \right)
\]  
(11)

All the utilized proposals will be symmetrical in this aspect.

### 3.3 Fixed-Sequence Simulations

The obvious choice of space when doing simulations of proteins is the space of conformations. We will denote the vector with coordinates that describe the conformation as \(r\). This could either be the set of atom positions, or the \(2N\) torsional angles. It is known from statistical mechanics that, at equilibrium, the proteins are distributed according to the Boltzmann distribution.

\[
P(r) = \frac{1}{Z} e^{-\beta E(r)}
\]  
(12)

where \(\beta = 1/k_B T\), and the partition function is given by
\[ Z = \sum_r e^{-\beta E(r)} \]  
(13)

We stress that the partition function is unique for each sequence, \( Z = Z(\sigma) \).

This gives the acceptance probability for the Metropolis step as
\[ A(r \to r') = \min \left( 1, e^{-\beta \Delta E} \right), \]  
(14)

with \( \Delta E = E(r') - E(r) \).

### 3.3.1 Monte Carlo Chain Updates

We use two methods, both common choices for the model, to propose chain updates. When a proposal is to be made the type of update is chosen with equal probability for both kinds of update.

The first type is a simple pivot update, where a single torsional angle is selected and assigned a new value.

The second type of update is a so called biased gaussian step (BGS). This update aims to make a local deformation of the chain. The outlying parts of the chain should undergo only minimal deformations [10].

### 3.4 The Multisequence Method

There are several ways to improve the functionality of the naive Metropolis algorithm. A common method is to expand the ensemble that is simulated beyond the Boltzmann distribution in equation 12. For example, in simulated tempering the temperature is allowed to vary [11]. In this thesis, we will focus our attention on the implementation and investigation of a method presented in [12]. In this hitherto little used method, the ensemble is expanded by allowing the sequence to vary. The sequence can be written as a vector, \( \sigma = (\sigma_1, \ldots, \sigma_N) \), where the \( \sigma_i \) are the amino acid types (h, p or G). Following reference [12], we sample the distribution
\[ P(r, \sigma) = \frac{1}{\Xi} e^{-\beta E(r, \sigma) + g(\sigma)}, \]  
(15)

with
\[ \Xi = \sum_{r,\sigma} e^{-\beta E(r) + g(\sigma)}, \]  
(16)

where \( g(\sigma) \) are free parameters.

This distribution gives the correct single sequence distribution, since the conditional probability is given by
\[ P(r|\sigma) = \frac{P(r, \sigma)}{P(\sigma)} = \frac{e^{-\beta E(r, \sigma) + g(\sigma)}}{e^{g(\sigma)} \sum_{r'} e^{-\beta E(r', \sigma)}} = \frac{1}{Z(\sigma)} e^{-\beta E(r, \sigma)}, \]

For updates of the protein structure, the acceptance probability remains the same as in the fixed-sequence simulations. With symmetrical sequence updates, the acceptance probability for moves in sequence space becomes

\[ A(\sigma \rightarrow \sigma') = \min \left( 1, e^{-\beta \Delta E + \Delta g} \right), \]

with \( \Delta E = E(r, \sigma') - E(r, \sigma) \) and \( \Delta g = g(\sigma') - g(\sigma) \).

### 3.4.1 The Parameters \( g(\sigma) \)

The parameters \( g(\sigma) \) can be chosen freely to give a reasonable distribution in sequence space. The marginal distribution becomes

\[ P(\sigma) = \sum_r P(r, \sigma) = \frac{1}{\Xi} e^{g(\sigma)} \sum_r e^{-\beta E(r, \sigma)} = \frac{1}{\Xi} e^{\beta F(\sigma)} e^{-\beta F(\sigma)}, \]

where we have used the identity \( Z(\sigma) = e^{-\beta F(\sigma)} \). By choosing

\[ g(\sigma) = F(\sigma)/k_B T, \]

we get a flat distribution in sequence space [13].

The choice given by equation 24 can be generated iteratively. Using equation 23, and taking the logarithm of both sides, we may use the fact that \( F(\sigma) \) is independent of the sampled distribution. This allows us to write

\[ -\beta F(\sigma) = \log P(\sigma) - g(\sigma) + \log \Xi \]

where \( \log \) denotes the natural logarithm.

Suppose now that the multisequence method has been used to sample a probability distribution \( \tilde{P}(\sigma) \), with parameters \( \tilde{g}(\sigma) \). We wish to determine the parameters \( g(\sigma) \) which give a flat distribution. By using equation 25 for both the prior and the posterior distributions, we get the relation
\[
\log P(\sigma) - \log \tilde{P}(\sigma) = \tilde{g}(\sigma) - g(\sigma)
\] (26)

However, since \( P(\sigma) \) is a constant, and therefore unimportant, we can choose

\[
g(\sigma) = \tilde{g}(\sigma) - \log \tilde{P}(\sigma)
\] (27)

to get a flat distribution.

Once a flat distribution has been achieved, we can also use equation 26 to find suitable parameters for an arbitrary distribution. In that case we can ignore the constant value \( \log \tilde{P}(\sigma) \), and the relevant update becomes (for convenience, we have removed the tildes)

\[
g(\sigma) \to g(\sigma) + \log P(\sigma)
\] (28)

where \( P(\sigma) \) is now the desired probability distribution. This scheme can be useful for example if we want to use the algorithm to speed up the simulation of a single sequence. While the choice in equation 24 has been used before (e.g. in [13]), this scheme is new as far as we know.

### 3.4.2 Sequence Updates

Updates in configuration space are done by the same procedure as for the fixed-sequence simulations. When making proposals for updates in sequence space we utilize two different update schemes. Note however that these are not both used during a single simulation run. Instead the type of update is chosen to control the size of the sequence space being visited.

The first type of update is to simply flip a single amino acid. This assures that the whole of sequence space can be reached, while making the process of selecting an update simple.

For longer sequences, it might be unwieldy to explore and collect statistics for every single sequence. Therefore, we utilize a second update type to flip two amino acids simultaneously. These amino acids are flipped in a coordinated way, so that the Hamming distance between any two visited sequences obeys \( \Delta H \geq 2 \). The method reduces the number \( n(\sigma) \) of visited sequences by a factor three.

To accomplish this, we identify each type of amino acid with a number between zero and two. For example, one can use \( n(h) = 0 \), \( n(p) = 1 \) and \( n(G) = 2 \). We then demand that the quantity

\[
S = \left( \sum_{i=1}^{N} n(\sigma_i) \right) \mod 3
\] (29)

is conserved when the sequence is updated.

Clearly, imposing this restriction will ensure that no two sequences with \( \Delta H = 1 \) can be visited. In addition we note that choosing all but one of the amino acids freely will uniquely determine the type of the final amino acid, through
\[
n(\sigma_j) = \left( S - \sum_{i \neq j} n(\sigma_i) \right) \mod 3 \quad (30)
\]

Thus the total number of allowed amino acids are decreased by a factor of three. An illustration of this can be seen in figure 3.

Updates which conserve the value of the sum given above can be realized through the following procedure.

1. Two amino acid positions are chosen.
2. A new amino acid type is assigned to the first position.
3. The type of amino acid of the second position is now uniquely determined by the requirement that \( S \) is conserved, and can be calculated using equation 30.

Note that the function \( n(\sigma_i) \) could have been chosen independently for each amino acid, as long as the values for different amino acids are unique. However, the present choice makes the updates particularly simple.

Figure 3: An illustration of the double-flip method for the simple case with only two amino-acids. Each dot represents a protein sequence, with the position determined by the values \( \sigma_i \). The colored lines mark the sequences with the same value of the sum in equation 29. Note that the \( \sigma_1\sigma_2 \)-plane is cyclic.

### 3.5 Observables

In order to compare results from different simulations, as well as data for different sequences, we use a couple of different observables.

For simple comparisons we use the end-to-end distance, \( R_{ee} \), defined by

\[
R_{ee} = |r_1 - r_N| \quad (31)
\]
Here, $r_1$ ($r_N$) is the position of the side-chain atom of the first (last) amino acid. To compare the results of different simulations, we will either compare the probability distribution of $R_{ee}$, or simply measure the average value, $\langle R_{ee} \rangle$ of this observable. In the latter case we will often be interested in the deviation in this estimate,

$$\delta R_{ee} = \langle R_{ee} \rangle - \langle R_{ee} \rangle_{eq}$$

(32)

where $\langle R_{ee} \rangle_{eq}$ denotes the equilibrium average, as measured by a longer simulation. We will also be interested in the observed standard deviation of the end-to-end distance, $\sigma_{R_{ee}}$ in simulations of a certain length.

When classifying the neutral nets of a specific structure, the end-to-end distance is clearly insufficient. For this reason we instead classify structures by the number of native hydrogen bonds they contain. However, since the structure at any finite temperature includes some degree of flexibility, this classification is done in several steps.

First, the existing hydrogen bonds are checked. A hydrogen bond is considered to exist if the NH and CO groups are within a certain cutoff distance. These are then compared with the hydrogen bonds present in the native state. A protein is considered to populate the native state if at least 75% of the native bonds are present.

To determine if a specific sequence folds to the native state the extent to which it populates the native state as defined above is determined. If the sequence folds to the same native state at least 70% of the time, that state is considered the native state of the protein. The choice of cut-off is somewhat arbitrary, and largely reflects the value necessary to achieve a reasonable size for both the neutral nets.

### 3.6 Simulations

The simulations can be divided in three parts: validation of the multisequence implementation, comparison of the performance of the multisequence and the fixed-sequence methods, and investigation of the neutral nets for model proteins of length eight amino acids.

Suitable temperatures for both chain-lengths were determined by comparing the average energy at different temperatures. The aim was to choose a temperature where the sequences investigated would fold, but not one where they would be frozen in the ground state. Note that all temperatures given are in model units with the Boltzmann constant set to unity, $k_B = 1$. Reasonable values of the parameters $g(\sigma)$ were determined by iterating equation 27 until a suitably flat distribution was achieved. If a non-even distribution was sought, the values were updated according to equation 28.

Data was collected once every 100 Monte Carlo steps, or every Monte Carlo cycle (MC cycle). Typically each simulation lasted $10^6$ MC cycles, as this was found to be a suitable time in order to collect enough statistics. To determine the
statistical certainty of the results, every simulation was performed independently until the standard errors were deemed sufficiently small.

The validation was performed by running simulations of two different six-amino acid sequences with both fixed sequence and with a flat distribution over all sequences. To ensure consistency, the probability distribution of the end-to-end distance in the two simulations were compared. All simulations were performed at $T = 0.27$, and each simulation was repeated five times.

In comparing the performance of the multisequence algorithm with the fixed-sequence version we focused on estimating the gain when simulating a single sequence. To this end we compared the precision in an estimate of $\langle R_{ee}\rangle$, for six different sequences.

Comparisons were made between three different strategies for the sampling of sequence distributions:

1. Fixed sequence: The naive Monte Carlo simulation.
2. Target-sequence: The multisequence method, with a marginal probability of (approximately) 50% for visits to the chosen target sequence. The distribution over the remaining sequences was flat.
3. Flat distribution: The multisequence with an (approximately) flat distribution over all sequences.

We are interested in studying both the statistical precision of the various strategies, and their effects on the thermalization time. The thermalization can roughly speaking be described as the time it takes for the system to reach the equilibrium distribution, given random initial conditions. To determine these two quantities, the value of $\langle R_{ee}\rangle$ was computed based on the simulated results of the first 500 000 MC cycles, as well as the next 500 000 MC cycles. Each simulation was repeated 25 times. For comparison, the simulations were compared with the estimated value from a $10^7$-MC cycles run with the flat-distribution multisequence algorithm.

For the investigation of neutral nets, the flat-distribution multisequence algorithm with double flips were used to simulate proteins with a length of eight amino acids. The choice of length was based on the fact that $\alpha$-helical structures are rather unstable in the 6-amino acid case, whereas longer sequences require a considerably larger simulation time. The native fold of each sequence was then determined using the method described in the preceding section. Fold switches where a single double-flip was sufficient to change the native conformation were also identified. All simulations were performed at $T = 0.32$, which was found to be a suitable temperature in some preliminary test runs. Five independent runs were performed to establish statistics.
4 Results

4.1 Validation of the Multisequence Algorithm

Our first task is to test the soundness of the newly implemented multisequence algorithm. Simulations of two different protein sequences were used to compare the results with the fixed-sequence algorithm. The sequences were chosen based on fundamental design principles. The choice was made with the aim of using one sequence natively folding to an α-helix, and one folding to a β-hairpin. This objective was not entirely achieved, as the designed α-helix most of the time

(a) Sequence hpGGph
(b) Sequence hpphhp

Figure 4: The minimum energy conformations for two different sequences as sampled in fixed-sequence simulations with a duration of $10^6$ MC cycles. Pictures generated using Jmol [14].

![Figure 4: The minimum energy conformations for two different sequences as sampled in fixed-sequence simulations with a duration of $10^6$ MC cycles. Pictures generated using Jmol [14].](image)

![Figure 5: The temperature dependence of the mean energy of the protein hp- phhp, as sampled using fixed-sequence simulations. Three independent runs were performed at each temperature. Also note that the error bars are so small that they cannot be seen in the figure except for the lowest temperatures.](image)
folded to a helix with greater radius. The two sequences still exhibit different thermodynamic behavior as evidenced e.g. by their ground states, shown in figure 4.

In order to find a suitable temperature, the behavior of one of the sequences were sampled at different temperatures, as can be seen in figure 5. At low temperatures, the sampling appears to be inaccurate. This is likely due to the fact that moves upwards in energy becomes increasingly rare, so the sampling simply finds a local minimum close to the initial conformation. These sampling problems make it rather difficult to choose a good temperature, but $T = 0.27$ was deemed a suitable choice.

Before starting simulations, the values of the parameters $g(\sigma)$ also had to be
determined. The probability distribution after a couple of iterations of using equation 27 can be seen in figure 6. While a completely uniform distribution has not been achieved, the sampling frequency for all of the sequences should be high enough to ensure that statistics are reliable.

As expected, the multisequence algorithm produced results consistent with the fixed sequence simulations, as can be seen in figure 7. As can be seen from these figures, the multisequence does, however, give equal or slightly lower statistical errors than simulations with a single sequence. In the following we will further explore this issue, as well as discuss what choice of parameters $g(\sigma)$ result in the most efficient sampling.

4.2 Computational Efficiency of the Multisequence Algorithm

Having verified that our new algorithm gives the correct results, we now turn to the question of computational efficiency. To do this we compare the precision in calculations of $\langle R_{ee} \rangle$ using the naive Monte Carlo on one hand, and the multisequence with and without a target sequence on the other. (For a longer description of the sampling methods, refer to section 3.6).

We use the same two sequences as before, as well as four additional sequences. Two of these were chosen in the same way as before, one to fold to a $\beta$-hairpin and one to fold to an $\alpha$-helix. The last two were generated randomly. All of the sequences are listed in table 1.

Table 1: The sequences used for comparing the computational efficiency of the different methods.

<table>
<thead>
<tr>
<th>Design</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$-helix</td>
<td>hpphhp</td>
</tr>
<tr>
<td>$\alpha$-helix</td>
<td>hpghhp</td>
</tr>
<tr>
<td>$\beta$-hairpin</td>
<td>hpGGph</td>
</tr>
<tr>
<td>$\beta$-hairpin</td>
<td>phGGhp</td>
</tr>
<tr>
<td>random</td>
<td>GphGpp</td>
</tr>
<tr>
<td>random</td>
<td>GhhphG</td>
</tr>
</tbody>
</table>

As was the case with the sequence hpphhp, the sequence hpghhp does not have a completely $\alpha$-helical behavior. It folds into three different folds with different frequencies. These folds are reminiscent of a “true” $\alpha$-helix, a helical shape quite similar to the energy minimum of the hpphhp sequence, and a $\beta$-hairpin. Although no quantitative measurement of this behavior has been made, the helical shapes seem to be more common than the hairpin. It should be stressed that the native state is of no real importance, for determining computational efficiency it is mainly the complexity of the free-energy landscape that is of importance.

In order to quantify the improvement of the multisequence algorithm, we consider two different computational problems. First, we study how the length and severity of the thermalization period with the different types of sampling. This
Figure 8: The average value of the end-to-end distance as estimated using the first $5 \cdot 10^5$ MC cycles of simulations with three different sampling strategies in sequence space at $T = 0.27$, with standard errors. Statistics were based on 15 independent runs. Note that the singlesequence simulations have a significant error due to thermalization in the two $\alpha$-helical sequences. The equilibrium line was determined using longer simulations.

Figure 9: The average value of the end-to-end distance as estimated using the final $5 \cdot 10^5$ MC cycles in a simulation of $10^6$ cycles with three different sampling strategies in sequence space at $T = 0.27$, with standard errors. Note that the thermalization process appears to have ended for all of the methods. The equilibrium line was determined using longer simulations.

is done by using the data from the first and second halves of the simulation. The results hereof can be seen in figures 8 and 9. To determine the errors, we compare the results with a simulation of $10^7$ MC cycles using a flat distribution in sequence space. Unfortunately, this is not enough to eliminate the effects of
statistical errors completely.

As expected the results are in better accordance during the second half of the run for all but a few data points. The exceptions are presumably the result of statistical fluctuations. The fixed-sequence simulations especially give a sizable error during the first half of the simulations for both the $\alpha$-helical sequences.

The second point of comparison is in determining the statistical errors in the different methods. Based on the results concerning thermalization times we choose to use only data from the second halves of the runs when calculating these errors. The results of this estimation can be seen in figure 10. We note that the two multisequence-based methods perform at least as good, and in some cases significantly better, than the fixed-sequence algorithm.

In order to establish the cause of the variable efficiency, we study the runtime history of the end-to-end distance, shown in figure 11. We note that the most troublesome of the $\alpha$-helical sequences (hpphhp) can be characterized as having two main states. Transitions between these states are very rare in the fixed-sequence simulations. In comparison, the multisequence method with a designated target sequence shows quite frequent transitions which results in more uncorrelated samples. We can also see the cause of the large thermalization period in figure 11b. Because of the small number of transitions the fixed-sequence algorithm sometimes gets stuck in the more uncommon state for quite a long time. Since the probability of collapsing to the two states from a random coil are not correlated to the equilibrium probability of finding them in these states, we get too many samples in the higher-$R_{ee}$ state for a long period of time.
Figure 11: The runtime sampling of a couple of different runs with fixed-sequence, and both varieties of multisequence runs. The runs are in general chosen to be representative, although the fixed-sequence run shown in 11b shows a particularly troublesome feature of that sequence (which is common although probably not representative). The first three subfigures show hard-to-sample \( \alpha \)-helical sequences, while the final one shows an easily sampled \( \beta \)-hairpin structure. Sample points are taken every 1000 MC cycles and are only plotted if the sequence of interest is visited. Note that this means that only one in 1000 actual visits are recorded, and thus the flat-distribution multisequence method gets very few points (we chose runs with lots of visits).

It is certainly of interest to compare with other sequences. We note that the sequence hhpphp has three different main states, but a higher transition probability between at least two of them. This of course reduces the thermalization period, but the higher number of states seems to result in an equilibrium distribution which is more difficult to sample (as seen in figure 10). The sequence hpGGph can be seen to have a very simple structure, with only a single state visited. As can be seen in figure 11d this gives the fixed-sequence algorithm no problems, and thus the multisequence methods no longer give any significant advantages.

We note that one of the random sequences (GhhphG) actually gives better statistics using the fixed-sequence algorithm. This is likely due to the fact that both ends consists of glycine amino acids. This should give the ends a large freedom of movement and thus a very broad distribution of the end-to-end distance. This negates the improvements from the multisequence since passage
between different “states” is extremely easy anyway. In addition, the need for
many sampled points increases, since the end-to-end distance is not precisely
defined.

4.3 Sequence-Structure Space of 8 Amino Acid Model
Proteins

One of the main advantages of the multisequence algorithm is that it can be
used to simulate a large number of sequences at the same time. To demonstrate
this ability we take the opportunity to do a more complete investigation of
the sequence-structure relationships for model proteins with length eight amino
acids. Because of the size of the sequence-structure space, we use double se-
quence flips (see section 3.4.2).

First off we need to adjust the temperature to a value suitable for proteins
of length 8 amino acids. Figure 12 shows the temperature dependence of the
average energy for a single eight-amino acid protein. The data from this graph
should make it clear that simulations at the temperature $T = 0.32$ is a reasonable
choice.

![Figure 12: The temperature dependence of the mean energy of the protein hpphhpph, as sampled using fixed-sequence simulations. Three independent runs were performed at each temperature. Note that the transition from an ordered to a disordered state occurs at around $T = 0.45$. Also note that the error bars are so small that they cannot be seen in the figure except for the lowest temperature.](image)

We use the data gathered to collect information about the native folds of each
protein, thereby mapping the neutral nets of both the $\alpha$-helix and the $\beta$-hairpin.
A graphical representation of the two nets and the fold-switches between them
can be seen in figure 13. We note that the net of the $\beta$-hairpin is about 4-5
times the size of the helix-net. In addition, the neutral sets of both structures are
connected, in the sense that transitions between any two sequences completely
It can also be seen that flips between the two nets are very common. Almost all proteins with a helical native structure can perform a fold-switch, as can approximately one third of the sequences in the β-net.

Table 2: The prototype sequences for the α- and β-nets.

<table>
<thead>
<tr>
<th>Conformation</th>
<th>Prototype sequence</th>
<th># of contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>α-helix</td>
<td>hpphhphp</td>
<td>12</td>
</tr>
<tr>
<td>β-hairpin</td>
<td>hGpGGpGh</td>
<td>26</td>
</tr>
</tbody>
</table>

As shown in figure 14, the sequences of the β-hairpin net show the same kind of relationship between mutational and thermodynamic stability as was seen for lattice proteins, described in sections 2.2.1 and 2.3. The correlation for α-helical sequences, shown in figure 15, is considerably less clear-cut. This might however, at least in part, be due to the lower number of sequences in the α-helical net.

Table 2 shows the designing sequences for each neutral net. They both correspond well with the choices used for six-amino acid proteins, and as such the results are not surprising. From studying the full sets of sequences we can note that the α-helical sequences are very sensitive to the inclusion of glycine, with only two of the sampled sequences including any at all. The β-hairpins have a much greater flexibility. We do note however that the existence of at least one glycine in positions 4 and 5 is necessary, whereas glycine amino acids at positions 1, 3, 6 and 8 is quite destabilizing. This might in part be due to the model formulation however, since hydrogen bonds involving glycine is less energetically favorable.
Figure 13: Schematic view of the $\alpha$-helix and $\beta$-hairpin neutral nets of proteins of eight amino acids length. Each node represents a specific protein sequence, with blue nodes folding to hairpins, and red ones forming helices. The edges connect proteins with $\Delta H = 2$ and are colour-coded according to the colour of their endpoints. Note that most sequences fold to neither conformation, and as such are not included. Green edges are used to highlight fold-switches. Graph generated using Gephi [15].
Figure 14: The relation between mutational and thermodynamical stability for proteins in the β-helical neutral net. The correlation coefficient is estimated to be $r = -0.6787$, which indicates that a clear correlation between the two sets exists.

Figure 15: The relation between mutational and thermodynamical stability for proteins in the α-helical neutral net. Due to the small number of points it is difficult to be certain of any trends and relationship between the two quantities. The correlation coefficient is estimated to be $r = -0.2359$. 
5 Discussion

The results above clearly shows that the multisequence algorithm gives significantly lower thermalization times and standard deviations for at least some sequences. We also note that the sequences for which the improvement is most pronounced are those with complicated free-energy landscapes. This means that the largest improvements can be seen for sequences which are difficult to simulate. It should also be noted that complicated free-energy landscapes are typical for real proteins, so the method may well give considerable advantages when simulating these.

Although we have no data to support this hypothesis, the fact that the multisequence algorithm increases the number of transitions between different states makes it reasonable to assume that it would also give a larger improvement for simulations at low temperature.

While the multisequence method does not provide large improvements for several of the sequences tested, it should be noted however that this comparison is rather unfair to the multisequence with a flat distribution. Since that method is able to sample the thermodynamic behavior for all six amino-acid sequences at the same time, the required simulation time is reduced by a factor $3^6 = 729$. When taking this into account it is obvious that the multisequence provides a considerable advantage if a large number of sequences are to be sampled.

The small differences between the two variants of the multisequence algorithms are quite interesting, as it seems to imply that the marginal probability of a sequence typically does not affect the simulation times. There could conceivably exist a local minimum somewhere between the two variants, but since they are very close in performance the gain would probably not be too large. One factor which has not been tested for is the distribution of the sequences which are not targeted. It is possible that it would be preferable to sample structures similar to the target sequence with a higher frequency than sequences with a large Hamming distance to the same.

While the improvements found are striking, it should be emphasized that the model used has several attributes which make it especially well suited for use along with the multisequence method.

First, the model has only three different amino acids, which make the sequence space comparably small. In a model which represents all 21 amino acids found in nature, exploration of the complete space is entirely unfeasible. If only a single sequence is to be sampled it should be possible to restrict the sequence space in some way however.

Second, the geometrical formulation of the model is such that the total number of atoms remain constant for all proteins of the same length. This makes the sequence flip considerably less technically challenging. In addition, the positions of the vast number of atoms need not be adjusted during flips. (There is one exception: the position of the sidechain molecule needs to be adjusted if the flip creates or destroys a glycine amino acid.)

We also note that the time spent generating suitable choices for the parameters $g(\sigma)$ is not included in the comparison above. This does take up a significant
part of the simulation time, and if the method is used to simulate a single sequence it might well be important to take this time requirement into account. If it is used to sample a large number of sequences it should still be obvious that the method does give a significant advantage.

It should be noted that there exists several other schemes to improve the performance of the naive Metropolis algorithm. We have not, however, done any comparisons of the performance of the multisequence algorithm with the performance of these other methods.

When looking at the results from investigations of the neutral nets, it should be noted that we have only simulated a third of all the possible sequences. This means that the specified designing sequences is not necessarily the real ones. It could also conceivably mean that some of the identified fold-switches in fact have an intermediate sequence which folds to neither native state. Since there are in fact two different intermediate states for each sequence flip, however, it seems unlikely that this amounts to a significant number of false fold-switches. In fact, a lot of the identified fold-switches would probably be two if flips were performed on single amino acids.

It should be noted that the statistical certainty on a few of the sequences are low enough that their designation as being part or not being part of a neutral net could be erroneous. This number should be low however, and should not significantly affect the conclusions drawn. It could also be argued that the cutoff is rather arbitrary, and whether a sequence folds in a certain way to within a fraction of a percent is hardly important.

It is interesting to note that the neutral net of the $\alpha$-helix is the smaller one. This result is not entirely unexpected however, as the short chain-lengths mean that each amino acid can form a hydrogen bond with only one other amino acid. In a longer chain most amino acids will be able to form two different bonds, resulting in a more energetically favorable state.

The comparably large number of possible fold switches suggests that, if the model is representative of real proteins in this aspect, these switches could be useful paths in the evolution of proteins. It should, however, be noted that no conclusive results can be drawn from this result, since both the model and the size of the proteins might have an influence on this behavior. For instance, a larger protein is likely to have a larger neutral net, and thus a comparably smaller “surface” which is susceptible to fold-switches.

The correlation between thermodynamic and mutational stability in the $\beta$-net follows the same pattern as has been seen in studies of lattice proteins. We again note that the measurement of the two types of stability are not equivalent.

Further investigations of the neutral nets should probably start with an expansion of the sampled sequences, either by running simulations with flips of only a single amino acid, or by calculating them through histogram re-weighting techniques [16]. Another plausible line of inquiry would be to investigate the existence of other structures than the two categorized in this thesis. Of course one could also try sampling longer sequences. This should be possible to do with the use of a sufficiently powerful computer.
6 Conclusions

The implemented multisequence algorithm was found to give a significant improvement compared to the naive Metropolis algorithm. We found that an advantage is to be had even if the multisequence algorithm is used to simulate a single six amino acid model protein. Thus, the method shows great promise of offering a fast and reliable way to simulate the thermodynamic behaviour of a large number of different sequences at the same time.

The subsequent simulations showed that the eight amino acid model proteins included both α-helical and β-hairpin like structures. The neutral net of the β-hairpin turned out to be larger than that of the α-helix. A considerable number of double mutation fold-switches between the two nets were found.
References


