Protein Fold Recognition by Assembly of Fragments

Emil Olovsson *

January 12, 2001

Supervisor: Arne Elofsson, Stockholm Bioinformatics Center,
Stockholm University

Examiner: Stefan Ståhl, Biochemistry department,
The Royal Institute of Technology, Stockholm (KTH)

*Stockholm Bioinformatics Center, Stockholm University, SE-106 91 Stockholm, Sweden
E-mail: emil.olovsson@mail.bip.net

Abstract

This study explores the ability of a combined exhaustive search and Metropolis annealing procedure to assemble native-like structures from fragments of unrelated protein structures. The simulated annealing procedure forms native-like structures for small α-proteins. The performance for β-proteins was poor. A simple scoring function based on statistics from the protein database was tried without success. The effect of fragment quality on the success of the procedure is explored. Although proper β-sheet forming did not occur, promising structures with β-strands was formed. The results suggest that this could be a useful method for ab initio protein folding together with a suitable scoring function.

Keywords: ab initio protein structure prediction; fragment assembly; computer simulation; multiple sequence alignment

Contents

1 Introduction

During the last decade there has been substantial progress in the development of algorithms for ab initio protein folding (see section 1.6.1). Due to great difficulties, ab initio folding algorithms are not likely to become useful methods of structure prediction for any but the smallest proteins for quite some time. However, such efforts are of crucial importance because they highlight, as few other experiments can, the challenges facing current studies of protein folding. [SKHB97]

1.1 The aim of this study

The aim of this study is to develop a method for constructing protein structures for proteins without known structure, but with known amino acid composition. In this study, no attempt is made to pick out the native-like conformations.

The idea is that more native-like proteins will be obtained by using information about the secondary structure. There have been a number of reports in which the secondary structure was kept fixed at that of the native structure [MFH94] [STD95].

Fidelis et. al. [FSBM94] suggested that building structures from fragments was likely to be unsuccessful because of the structural divergence of local fragments and the weakness of local sequence-structural correlation. Simons et. al. [SKHB97] however states that the relative success of methods similar to the one used in this work, may be due to two reasons:

1. Reasonable tertiary structure can be assembled without near perfect local structural agreement between simulated and native structures.

2. Even weak local sequence biases can significantly affect the likelihood of generating different tertiary structures.

1.2 Protein structure

The most significant property of all proteins is that they, in contrast to all synthetic polymers, form a well-defined three-dimensional structure.

1.3 Protein function

Proteins are the main component of all cellular life, and the task to understand protein function is the main challenge in biochemistry at present.

Proteins are polymers of 20 different building blocks, called amino acids. The amino acids are carefully chosen by nature to have different chemical properties, in this way different molecules with vary different properties can be built with few building blocks. Proteins are, in contrast to all synthetic polymers (plastics) monodisperse; i.e. all molecules have the same structure and molecular weight. This is important when you want the protein to perform a specific task, and nothing else. This amazing correctness by the cells in building proteins makes proteins an attractive target for a big variety of tasks.

Protein function are very much dependent on its three-dimensional structure; for a sidechain to be able to interact with its environment, it must not be

buried inside the protein. This makes determining the structure of proteins an attractive task.

1.4 Cellular machinery

It is difficult to measure the arrangement of amino acids in the proteins. However, the sequence of nucleotides in DNA and RNA are easily determined.

The cellular machinery is basically the same in all living organisms. DNA is transcribed into mRNA, which is translated into proteins. In bacteria this process is performed in a highly predictable and straightforward way. It is thus possible to determine the sequence of the amino acids in the proteins just by knowing the sequence of the nucleotides in the DNA, provided that you know where the genes are located. In eucaryotes the situation is more complex due to non-coding regions within the genes (introns), and the task is not straightforward anymore. If the sequence of the mRNAs is determined instead, this problem can be circumvented. The mRNA has certain tags attached to them that makes it possible to distinguish mRNA from RNA. A problem, but also in some cases an advantage, with this is that only the expressed genes are detected.

1.5 Methods of determining protein structure experimentally

The investigation described in this paper, as well as all theoretical approaches to determine protein structure, depends heavily upon experimentally determined protein structures. Structures are measured using techniques such as X-ray crystallography and NMR.

1.6 Folding Simulations

The experimental way of determining protein structures is a time consuming operation, therefore different methods of predicting protein structure have been proposed.

The most successful of these is to find a protein with known structure with the same fold as the target protein, called a homolog. This is usually accomplished by aligning the target protein against all proteins with known structure. For examples of aligning methods see section 2.4 and 2.5.

Several groups have reported that it is possible to identify homologous folding proteins among a large number of folds [JTT92]. Even proteins that have no detectable sequence homology but have similar three-dimensional structure can be detected as having the same fold. Bowie et al. [BLE91] have detected a structural relation between the cAMP receptor protein and the cAMP-dependent protein kinase family, which have a sequence similarity well below 25 per cent. [Elo93]

1.6.1 Ab Initio Protein Folding

For proteins where no homolog with known structure is known, other methods have to be used. The methods trying to predict the sequence without using knowledge about homologs are often called ab initio protein-folding methods.

The main problem one addresses in ab initio methods, is the enormous number of structures possible given an amino acid sequence [Lev68]. To reduce the number of structures that have to be investigated, different approaches have been tried. One common method is to define a lattice, and let the atom positions be limited to this lattice.

Another important factor to take into account is to what level of detail the protein is represented. Ideally you want to describe it in as much detail as possible, but due to computational limitations this is often not possible. A common approach is to use only one or two atoms per residue, representing the α-carbon and a possible sidechain atom.

Ortiz et al [OKR+99] used PSI-BLAST [AMS+97] and PHD [RS93] to derive tertiary and secondary structure restraints. Using these restraints, a Monte Carlo search was run using a lattice model [KS96]. Finally the reduced models were converted into all-atom models using MODELLER [Sal93].

Samudrala et. al. [SXHL99] first tested all possible compact conformations on a simple tetrahedral lattice. A large subset of conformations were selected using a lattice-based scoring function, and detailed all-atom models were built using predicted secondary structure. These structures were then evaluated in two steps using scoring functions of increasing complexity.

This paper tries to use building blocks from proteins with known structure to build structures. The theory behind this is that nature tends to use the same templates when constructing proteins. Another reason is that different energy forces govern the local and the global packing of proteins. The local packing is mainly determined by local forces, such as preference of rotamers and packing of sidechains, while the global packing is mainly determined by hydrophobicity.

An approach similar to the one used in this study has been performed by Simons et. al. [SKHB97] with some success. Simons uses different criteria for generation of fragments and evolving of structures than this study. Simons method is to date the most successful method for ab initio protein structure prediction; this was demonstrated by Baker et. al. at CASP 4 (Critical assessment of methods of proteins structure predictions) [CAS00]. See [MHB+97] for a description about the CASP meeting.

2 Theory

2.1 Conversion between cartesian coordinates and dihedral angles

Given the Cartesian coordinates x_i, y_i and z_i of four covalently linked atoms (a_i, a_j, a_k, a_l), see figure 1, we can define the position of the fourth atom (a_l) relative to the three others (a_i, a_j, a_k) using the bond length ($|\overline{kl}|$), the bond angle ($\theta_{j,k,l}$) and the dihedral angle ($\phi_{i,j,k,l}$).

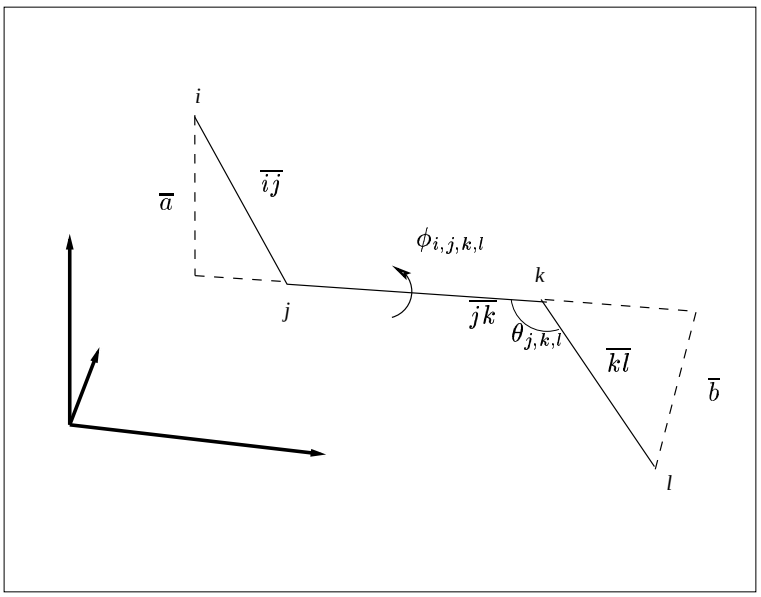

Figure 1: Vectors used in the calculation of the dihedral angle. The direction of the vectors ($\overline{ij}$, $\overline{jk}$ and $\overline{kl}$) are from left to right in the figure.

These quantities can be calculated in the following way, where the vectors are defined in figure 1.

The bond length $|\overline{kl}|$ is given by:

$$|\overline{kl}| = \sqrt{(x_l - x_k)^2 + (y_l - y_k)^2 + (z_l - z_k)^2} \tag{1}$$

The bond angle ($\theta_{j,k,l}$) can be calculated by:

$$\theta_{j,k,l} = \arccos\left(\frac{-\overline{jk}}{|\overline{jk}|} \cdot \frac{\overline{kl}}{|\overline{kl}|}\right) \tag{2}$$

The dihedral or torsion angles describe how the molecule is rotated around one specific bond. More specifically, the torsion angle is the angle you see between the two bonds neighbouring the bond in question, when looking from a

viewpoint in the bonds extension. The dihedral angle is calculated by using the dot product between the vectors $\overline{a}$ and $\overline{b}$, which are projections of the planes π_{ijk} and π_{jkl} in a plane orthogonal to $\overline{jk}$.

The vectors $\overline{a}$ and $\overline{b}$ can be calculated in the following way, where all the vectors are taken from figure 1.

$$\overline{a} = -\overline{ij} + (\overline{ij} \cdot \frac{\overline{jk}}{\overline{jk} \cdot \overline{jk}})\overline{jk} \tag{3}$$

$$\overline{b} = \overline{kl} - (\overline{kl} \cdot \frac{\overline{jk}}{\overline{jk} \cdot \overline{jk}})\overline{jk} \tag{4}$$

Then the dihedral angle ($\phi_{i,j,k,l}$) is given by:

$$\phi_{i,j,k,l} = \arccos(\frac{\overline{a}}{|\overline{a}|} \cdot \frac{\overline{b}}{|\overline{b}|}) \tag{5}$$

2.2 The structural arrangement of proteins

Proteins are polymers of 20 amino acids. Each of the amino acids consist of an amino group, a carboxyl group, a hydrogen atom, and a group called side chain, all of which are bonded to an α-carbon atom. This carbon atom is named α because it is adjacent to the carboxyl (acidic) group.

In proteins, the α-carboxyl group of the amino acid is joined to the α-amino group by a peptide bond (also called an amide bond). Two amino acids form a dipeptide by loss of a water molecule. The peptide bond contains a conjugated C-N bond. This means that the free electron pair, in the nitrogen atom, is "smeared out" over the nitrogen, the carbon and the oxygen atom.

The α-carboxyl groups, the α-carbon atoms and the α-amino groups of a protein is called the backbone. All atoms in the backbone, except the hydrogen and oxygen, are covalently bonded to one another in a series. This means that the structure of the backbone can be described with three parameters for each bond, namely the bond length, the bond angle and the torsion angle as described in section 2.1.

There are three different types of bonds in the backbone: C_α-C, conjugated C-N and N-C_α. The bond lengths in the backbone are quite constant in all residues. The C_α-C, C-N and N-C_α bonds have lengths of about 1.51, 1.32, 1.46 angstrom respectively. The bond length of the conjugated C-N bond (1.32 angstrom) is in between that of a C-N single bond (1.49 angstrom) and a C=N double bond (1.27 angstrom).

The bond angles in the backbone are also quite constant. The N-C_α-C bond have an angle close to the tetrahedral angle of 109.5°, while the C_α-C-N and the C-N-C_α bond have an angle of about 120°.

The torsion angles can be named ϕ, ω and ψ according to figure 2. As mentioned before, the carbonyl carbon-nitrogen bond is conjugated, which means that the π molecular orbital is partly occupied. This means that there is no rotation around this bond, and the ω angle is set to a value of 0°, which means that the peptide is always planar. [Str95]

The conclusion of this is that a good description of the backbone structure can be obtained from the ϕ and ψ angles alone. A pair of ϕ-ψ angles is often referred to as a rotamer.

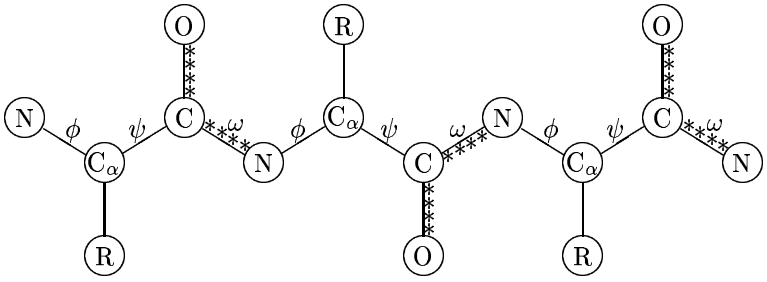

Figure 2: Schematical representation of the protein backbone. N means a Nitrogen atom, C means a Carbon atom, C_α means an α-carbon, O means an Oxygen and R Means a sidechain. The molecular π orbitals are marked with *.

2.3 The sidechains give the protein its properties

All proteins are built in the same way; the only thing that differs is the sidechains, which give the protein its properties. As mentioned above, proteins are polymers, but unlike many synthetic polymers, proteins are always unbranched. Proteins also fold into a compact structure, in contrast to most synthetic polymers. Cases where proteins do not form compact structures are often combined with disease.

Proteins often form complexes with DNA or RNA, but most often with other proteins. This is often referred to as the quaternary structure.

2.4 Alignments

Accurate alignments of sequences are needed for many types of analyses. Aligned sequences are the basis of phylogenetic analysis and of modelling of protein conformation. It can be used to identify functions of genes and proteins. Alignment methods are also used to search for similarities between new sequences and sequences in databases. Depending on the purposes, different properties of the alignment algorithm are important; searches in extensive databases require speed, while algorithms for alignments of homologous sequences can be optimised to use all available information to produce the most reliable alignment. [Elo00b]

2.4.1 Definition of an alignment

An alignment refers to the procedure of comparing two or more sequences by looking for a series of individual characters or character patterns that are in the same order in the sequences. Identical or similar characters are placed in the same column, and non- identical characters can either be placed in the same column as a mismatch or opposite to a gap in one of the other sequences. In an optimal alignment, non-identical characters and gaps are so placed to bring as many identical or similar characters as possible into vertical register. Two main

types of sequence alignment have been recognised, Global and local. The global alignment optimises the alignment over the full-length of the sequences. In local alignment, stretches of sequence with the highest density of matches are given the highest priority. The following is an example of global and local alignment.

Global alignment:

```
LGPSTKDFGKISESREFD
|      ||||    |
LNQLERSFGKINMRLEDA
```

The alignment is stretched over the entire sequence lengths to include as many matching amino acids as possible up to and including the sequence ends. Although there is an obvious region of identity in this example (the sequence FGKI), a global alignment may not align such regions in order to favour matching more amino acids along the entire sequence length.

Local alignment:

```
----------FGKI----------
          ||||
----------FGKI----------
```

Local alignment of the same sequences as above. In this case, the alignment tends to stop at the ends of regions of identity or strong similarity. A much higher priority is given to finding these local regions than to extending the alignment to include more neighbouring amino acid pairs. Dashes indicate sequence not included in the alignment. This type of alignment favours finding conserved amino acid motifs in related protein sequences.

2.4.2 Dynamic programming

Dynamic programming (DP) is one of the most frequently used computer algorithms in bioinformatics. It is used in most sequence search protocols as well as in most alignment problems. Often DP is used to find the "best" alignment between two strings (i.e. DNA or protein sequences). [Elo00a]

Dynamic programming solves problems by combining the solutions into subproblems. Dynamic programming can be used when a problem can be divided into subproblems and when these subproblems are not independent, i.e. when subproblems share subsubproblems.

DP is most frequently used to optimisation problems, for instance to find the best score when aligning two sequences. Each solution produce a certain score and we want to produce the optimal solution, i.e. the solution with the highest (or lowest) score.

DP algorithms can be divided into 4 steps:

1. Characterise the structure of an optimal solution.
2. Recursively define the value of an optimal solution.
3. Compute the value of an optimal solution in a bottom-up fashion.
4. Construct an optimal solution from computed information.

Step 4 can be omitted if only the value and not the exact details are needed. This is for instance used when searching a database, when we are not interested in the alignment but only in the score for a particular match.

2.4.3 Global Alignments (Needleman-Wunch)

Global alignments aim at optimally aligning two or more sequences. The most used methods for global alignments are based on algorithms originally developed by Needleman and Wunch and modified by Sellers. This procedure (the NWS method) is using a dynamic programming algorithm that simplifies the enormous task of calculate a score for all possible alignments of two sequences with gaps of any lengths. The sequences to be aligned are arranged as rows and columns of a rectangular matrix. A score is calculated for each position of the matrix according to three possible events: replacement (or conservation) of a residue, insertion in sequence A or insertion in sequence B. The matrix elements D_i,j are filled with numbers according the rule: [Elo00c]

$$D_{i,j} = \max \begin{cases} (D_{i-1,j-1} + \beta(A_2, B_1)) & \\ (D_{i,j-k} + w(k)) & k = 1, ..., j-1 \\ (D_{i-k,j} + w(k)) & k = 1, ..., i-1 \end{cases} \quad (6)$$

Here the first alternative corresponds to the case of no insertions or deletions: the previous diagonal element is added to the score β for the current pair of residues (A_i, B_j). The other two alternatives correspond to insertions in A or B: the element above or to the left is added to the score for a deletion of length k. The scoring begins at the first element of the matrix, corresponding to one end of the sequences. An example using a short nucleotide sequence is shown in the figure below. The total score is found as the last element in the matrix. The actual sequence alignment is obtained from the path through which this numbers are obtained.

2.4.4 Local Alignments (The Smith-Waterman algorithm)

The dynamic programming algorithm can also be used for finding local sequence similarities. The Smith and Waterman algorithm is very similar to the NWS method except that a calculated negative number for a matrix position is replaced by zero, indicating that no sequence similarity has been detected up to that point. When all the matrix elements have been calculated, the maximum number in the matrix is located, and the alignment is traced back from this point until the first positive number. [Elo00c]

2.5 BLAST: A simplification of Smith-Waterman

The BLAST algorithm [AGM$^+$90] uses a word-based heuristic to approximate a simplification of the Smith-Waterman algorithm known as the maximal segment pairs algorithm. Maximal segment pairs alignments do not allow gaps and are specified by an equation that includes only the first and fourth terms of the Smith-Waterman equation presented above. Maximal segment pair alignments have the very valuable property that their statistics are well understood [AG96]. Thus, we can readily compute a significance probability for a maximal segment

pair alignment. Recent advances in maximal segment pairs statistics allow the use of several independent segment alignments to be used in evaluating the significance probability.

The price for being able to readily compute a significance probability is that the alignments can not have gaps [AG96]. Thus, the evolutionary model requires that there be a fairly long stretch of sequence that has evolved without insertions or deletions, or at least with a complimentary pattern of insertions and deletions that does not significantly disrupt the alignment.

2.5.1 Psiblast

Position-Specific Iterated BLAST(psiblast) is a development of the gapped blast algorithm. Iterated profile search methods have led to biologically important observations but, for many years, were quite slow and generally did not provide precise means for evaluating the significance of their results. The principal design goals in developing the Position-Specific Iterated BLAST (PSI-BLAST) program [AMS+97] were speed, simplicity and automatic operation. The procedure PSI-BLAST uses can be summarised in five steps: [AK98]

1. PSI-BLAST takes as an input a single protein sequence and compares it to a protein database, using the gapped BLAST program [AMS+97].

2. The program constructs a multiple alignment, and then a profile, from any significant local alignments found. The original query sequence serves as a template for the multiple alignment and profile, whose lengths are identical to that of the query. Different numbers of sequences are aligned in different template positions.

3. The profile is compared to the protein database, again seeking local alignments. After a few minor modifications, the BLAST algorithm [AMS+97], [AGM+90] can be used for this directly.

4. PSI-BLAST estimates the statistical significance of the local alignments found. Because profile substitution scores are constructed to a fixed scale [KA90], and gap scores remain independent of position, the statistical theory and parameters for gapped BLAST alignments [AG96] remain applicable to profile alignments [AMS+97].

5. Finally, PSI-BLAST iterates, by returning to step (2), an arbitrary number of times or until convergence.

Profile-alignment statistics allow PSI-BLAST to proceed as a natural extension of BLAST; the results produced in iterative search steps are comparable to those produced from the first pass. Unlike most profile-based search methods, PSI-BLAST runs as one program, starting with a single protein sequence, and the intermediate steps of multiple alignment and profile construction are invisible to the user.

2.6 Monte Carlo Simulations

The Monte Carlo method creates an ensemble of structures similar to an ensemble created from a molecular dynamics simulation. When describing an

ensemble of conformations it is not correct to just generate a number of random conformations, because such an ensemble would contain too many conformations with a high potential energy [Edh90]. We want the structures to be distributed according to a Boltzmann distribution, where the number of structures with low potential energy is high. The easiest way to accomplish this is to let the next conformation in the iteration depend on the earlier conformation. This will generate a so-called Markov chain of conformations. [Elo93]

We have to define a transition probability q_{ij} that describes the probability for the system to be in conformation i at step n given that it was in conformation j at step n-1

$$q_{ij} = Prob(i, n|j, n-1) \tag{7}$$

We can describe the probability distribution P(i,n) as:

$$P(i,n) - P(i,n-1) = \sum_{j \neq i} q_{ij} P(j,n-1) - (\sum_{k \neq i} q_{ki}) P(i,n-1) \tag{8}$$

In order to obtain a Boltzman distribution, the transition probabilities should be calculated so that:

$$\lim_{n \to \infty} P(i,n) = \alpha e^{-\nu(i)/kT} \tag{9}$$

where ν(i) is the potential energy of conformation i and kT is a constant called simulation temperature in this report (k is Boltzmans constant and T is the temperature in molecular dynamics simulations for which the method was originally used). One way to obtain this is [MRRT53]:

$$q_{ij} = \frac{1}{\tau} \min(1, e^{-\frac{\nu(i)-\nu(j)}{kT}}) \tag{10}$$

where τ is 1 for a few chosen transitions and ∞ for the rest.

Protein	Class	Size
1ctf	$\alpha+\beta$	68
1r69	α	63
1sn3	$\alpha+\beta$	65
1ubq	$\alpha+\beta$	76
2cro	α	65
3icb	α	75
4pti	$\alpha+\beta$	58
4rxn	β	54

Table 1: The proteins used in this study [PHL97]. The protein coordinate files are available at the Protein Data Bank [GBSB00].

3 Methods

Below is a description of the algorithm used in this study to generate candidate structures for a set of known proteins. A flowchart over the events in the procedure is given in figure 4 The proteins studied in this study all have known tertiary structures and are listed in table 1. The proteins 2cro and 1r69 are homologs with an rmsd-value of 0.79 Ångström (see section 3.6.1. These proteins were chosen because they were previously studied in a study by Park et. al. [PHL97].

3.1 Generating fragments

Before the structures can be generated, we have to generate the fragments used to generate the structures. We identify, by some method, a fragment of a protein with known tertiary structure that has a similar fold as our target fragment. The selection process is sketched in figure 3.

We use fragments with a length of nine amino acids. In this study, three different methods to generate fragments are used.

1. Randomly chosen fragments from a subset of proteins. (Random method)
2. Sequence alignment using psiblast. (Psiblast method)
3. Pick the fragment with the lowest root mean square distance of the α-carbons. This requires that the structure of the protein investigated is known. (Rmsd method)

In the rmsd group, the fragments are selected from a subset of seven proteins. This small subset of proteins ensures that there are substantial differences between the fragments used in the simulation and the native fold. In the psiblast method a larger (319 proteins) subset of proteins was used.

Among the proteins where fragments were picked from, there was one case of homology. In protein 1ubq, method psiblast, the protein 1aar was included. These proteins have a sequence similarity of 100 percent and thus are said to be the same protein. All other proteins used had a sequence identity with the target protein lower than 25 percent.

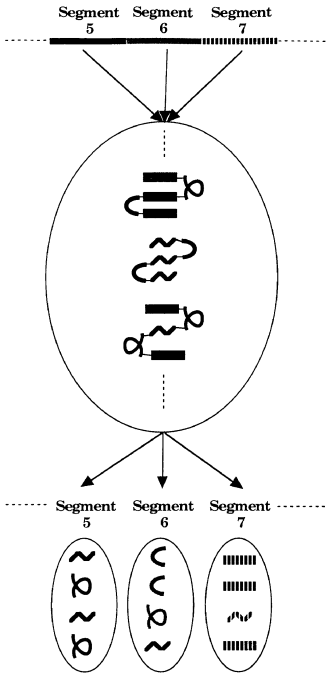

Figure 3: Schematical representation of the selection process of fragments. An amino acid chain is divided into several segments and segments of proteins that fit this segment are found. In this study, three different methods of finding segments are found. The picture is taken from [BE94]

In the random group the same subset as for the rmsd group was used. In table 2 the fraction of fragments with a rmsd of less than 1 Ångström obtained for the different methods is tabulated.

For a description of the psiblast algorithm see section 2.5.1.

The fragments are saved as a set of ϕ/ψ angles. A pair of ϕ/ψ angles will be referred to as a rotamer.

3.2 Assembly of fragments

At moderate resolution the backbone in a protein has constant bond- length and angles. The only thing that differs between different proteins is the torsion angles. The bond between the nitrogen and the carbonyl group is a conjugated double bond, and does not have freedom to rotate. Thus the structure of the backbone part of one amino acid can be described as a pair of ϕ/ψ angles. The fragments are thus described as a set of ϕ/ψ angles. If one angle in the middle of the protein is changed, the two halves in either side of the change will move with respect to the other. This will often result in a non-valid structure.

There are two suggested ways to deal with the problem described above. One is to make changes small enough not to affect the structure significantly. The other is to make a local move, i.e. you make a change at one place in the

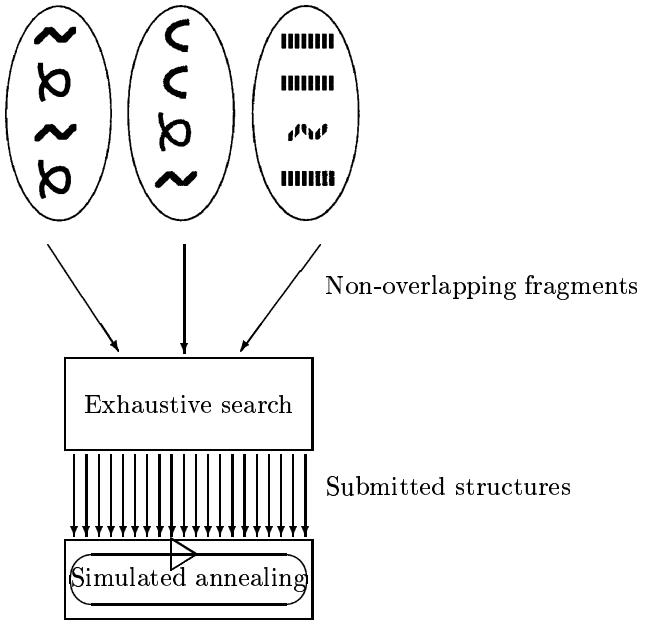

Figure 4: Flowchart of the protein building process. The fragments are fed into an exhaustive search where every combination of non-overlapping fragments are evaluated. The 100 best structures from each round is submitted to a monte carlo search where 1500 cycles of simulated annealing is performed for each structures.

structure, and then you make additional changes to ensure that the structure is not changed beyond a certain distance. [ELGD95]

In this study, local moves are not used. Therefore it is desirable to have similar fragments for each position. This condition is not met for the random sequences, which results in few accepted changes in the Metropolis search.

3.3 Search criteria

The basic assumption in this study is that the protein studied is compact and that the atoms of the protein do not collide. All proteins are certainly not compact; the anti-oncogene p53 could be mentioned as an example of a protein that does not fold into a compact state. The measure used for compactness is the gyration, which is defined as:

$$\sum (x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2 \tag{11}$$

where (x_i, y_i, z_i) are the coordinates of one α-carbon and $(\hat{x}, \hat{y}, \hat{z})$ are the coordinates of the centre of the protein i.e. the mean of all (x_i, y_i, z_i). To increase the speed, only alternate α-carbons were used.

The second search criterion is that there may not be any overlap between atoms. Two atoms are said to be overlapping when the distance between the centres of atoms is less than 2 Ångstrom. To simulate the sidechains, a centroid

Protein	Good	PsiBlast	Random
1ctf	0.53	0.27	0.01
1r69	0.65	0.26	0.00
1sn3	0.19	0.02	0.00
1ubq	0.43	0.19	0.00
2cro	0.66	0.25	0.00
3icb	0.53	0.24	0.00
4pti	0.21	0.04	0.00
4rxn	0.10	0.01	0.00
ALL	0.41	0.16	0.00

Table 2: Fraction of the fragments that are correct, i.e. that have an rmsd of less than 1Å

is added for each residue and all pairwise distances between α-carbon - α-carbon, α-carbon - centroid, centroid - centroid is calculated.

The centroids are constructed from the α-carbon coordinates (r_i values) in the following way. First, two unit vectors, $\vec{x}$ and $\vec{y}$, are calculated by the relations:

$$\vec{x} = \frac{(r_i - r_{i-1}) + (r_i - r_{i+1})}{|(r_i - r_{i-1}) + (r_i - r_{i+1})|} \tag{12}$$

$$\vec{y} = \frac{(r_i - r_{i-1}) \times (r_i - r_{i+1})}{|(r_i - r_{i-1}) \times (r_i - r_{i+1})|} \tag{13}$$

The centroid position, r_c, is calculated by the relation:

$$r_c = l\cos(\theta)\vec{x} + l\cos(\theta)\vec{y} \tag{14}$$

where l is the distance of the centroid from the α-carbon, and θ is the out-of-plane angle. The parameters were set to $\theta = 37.6°$ and l according to table 3.

3.4 Exhaustive search using non-overlapping fragments

The total number of possible conformations is $n_{frag}^{(n_{res}-n_{rot})}$ where n_{frag} is the number of fragments generated for each position, n_{res} is the number of amino acids in the protein, and n_{rot} is the number of rotamers within each fragment. For a protein with 60 amino acids there are 10^{50} possible conformations with fragment sets of 10 fragments containing 9 rotamers. These are not possible to search exhaustively, however there are $n_{frag}^{(n_{res}/n_{rot})}$ conformations of non-overlapping fragments if the fragments start every ninth residue. For the 60 amino acid protein as above, this gives 10^7 fragments. This amount is low enough to be examined exhaustively.

Let us consider an example with 9 fragments for each position and 7 different non-overlapping positions, (i.e. if the protein have between 55 and 63 residues)

	l
PHE	2.5
HIS	2.4
TRP	2.8
TYR	2.3
CYS	1.7
MET	2.6
THR	1.9
SER	1.9
ARG	2.4
LYS	2.1
GLN	2.4
ASN	2.5
GLU	2.0
ASP	1.8
PRO	1.9
ILE	2.0
LEU	1.5
VAL	1.5
ALA	1.5
GLY	0

Table 3: α-carbon - centroid distance

the non overlapping fragments are labelled from 0 to 8, and we denote a specific structure e.g. (1 2 3 4 5 6 7). Because of memory considerations, only a small subset of the structures is scanned at a time.

In this study 100 000 structures are scanned at a time, and only the 100 best according to the criteria described above are saved. For this selection process to work, ideally each subset of 100 000 structures should represent the complete set. The obvious way to do this would be to pick 100 000 structures at random, but that is not a very efficient way to do an exhaustive search. The second best way is to spread the subset over the complete set. In the example considered, this is achieved by increasing the loop variables by 8 for each step. The first subset will thus consist of (8 0 0 0 0 0 0) (7 8 0 0 0 0 0) (6 7 8 0 0 0 0)...

The gyration is much faster to compute than the overlap; therefore the gyration is first computed for all 100 000 structures. The list is then sorted with respect to gyration. The overlap is then evaluated for each structure, beginning with the one with the lowest (best) gyration, until 100 structures without overlap have been found.

3.5 Metropolis Monte Carlo search

The moderately compact, non-overlapping structures found using the algorithm described above are used as seeds for Metropolis Monte Carlo searches. For details about Metropolis Monte Carlo searches see section 2.6. The result of the Monte Carlo search relies on the simulation temperature. In this study a simulation temperature (kT value) of 15 is used. This value is dependent on the magnitude of the gyration, and thus the size of the protein, and is thus not

valid for all proteins. It seems reasonable for the proteins used in this study though. Since the number of Monte Carlo iterations run for each seed is small, this should not have a major impact on the result.

If we assume that the protein simulated has a helix, the entire helix should originate from the same fragment in the Monte Carlo search, or we will acquire very strange helices. Since our method of determining fragments in step 1 probably is better at recognising helices than loops, then the fragments describing a helix should be more closely resembling the native fragment than the fragments describing loops.

For each iteration step, replacing a fragment from the old structure creates a new structure. The likelihood of an exchange at a given position is proportional to the variance of that position. If the resulting structure lacks overlap and fulfils the Metropolis criterion (10), it is accepted to the next iteration. For every 30 iteration steps, the structure is saved if a new structure has been accepted during the last 30 iteration steps. In this study, 1500 iterations are performed for each seed.

3.6 Evaluation

3.6.1 Rmsd

The structures are evaluated using the root mean distance between the α-carbons in the correct structure and the acquired structure after the structures have been aligned on top of each other. This is shown in equation 15 with the two proteins denoted as A and B and n is the number of α-carbons in the atoms. This value is referred to as the rmsd value.

$$\frac{\sqrt{\sum_{i=1}^{n}(x_{A_i}-x_{B_i})^2+(y_{A_i}-y_{B_i})^2+(z_{A_i}-z_{B_i})^2}}{n} \tag{15}$$

3.6.2 Secondary structure preservation

The acquired structures are assigned a secondary structure using the stride algorithm (secondary STRuctural IDEntification) [FA95]. Stride aims at reproducing the criteria used by crystallographers to practically assign secondary structures in newly determined protein structures. Two main structural properties are considered, namely hydrogen bond patterns and backbone geometry, expressed as backbone torsion angles (ϕ/ψ). Each residue is classified as belonging to one of five states, H, E, T, G, or C, where "H" stands for α-helix, "E" for β-sheet (i.e. Extended), "T" for turn, "G" for 310-helix and "C" for coil. In this study, "H" and "G" are both categorised as α-helix and "T" and "C" as turns.

The secondary structure similarity is measured as the fraction of amino acids in the protein assigned to the same group of secondary structures in the acquired protein and in the native fold.

This definition of structural similarity may not be perfect, because it is hard to define exactly where a helix starts and ends. It is however a definition that is easy to evaluate automatically, and objectively.

Note that this definition is bound to assign moderate similarities although the proteins are not related at all since there are only three groups of secondary

structure.

Since this method tries to preserve the predicted secondary structure from the fragments, the emphasis here is that higher quality of the fragments does increase the secondary structure similarity.

3.6.3 Energy functions

To be able to detect near-native folds, we will have to have an energy function that estimates the energy of a given structure. The structures were evaluated using a function related to the vdw function described by Park et al [PL96].

The energy function is a distance dependent contact potential, similar to the van der Waals function. The sidechains are represented by a centroid that is calculated in the same way as described in section 3.3. All α-carbons are considered energetically equivalent, whereas the centroids have specific energies derived by statistics from the pdb database [GBSB00].

The energy is calculated in the following way:

$$E = \sum_{i=1}^{N}\sum_{j=i+4}^{N}(\frac{A_{ij}}{r_{ij}^8} - \frac{B_{ij}}{r_{ij}^4}) + \sum_{i=1}^{N}\sum_{j=i+4}^{N}(\frac{A_{\alpha\alpha}}{r_{\alpha_i\alpha_j}^8} - \frac{B_{\alpha\alpha}}{r_{\alpha_i\alpha_j}^4}) +$$

$$+\sum_{i=1}^{N}\sum_{j=1}^{i-3}(\frac{A_{i\alpha}}{r_{i\alpha_j}^8} - \frac{B_{i\alpha}}{r_{i\alpha_j}^4}) + \sum_{i=1}^{N}\sum_{j=1}^{i+3}(\frac{A_{i\alpha}}{r_{i\alpha_j}^8} - \frac{B_{i\alpha}}{r_{i\alpha_j}^4}) \quad (16)$$

where $A_{ij} = -\epsilon_{ij}(R_{ij}^a)^8$ and $B_{ij} = -2\epsilon_{ij}(R_{ij}^a)^4$

The values for ϵ_{ij} and R_{ij}^a used in this study are tabulated in table 8 and table 9 in the appendix.

Protein	100 saved structures				1000 saved structures			
	<7Å	<6Å	<5Å	<4Å	<7Å	<6Å	<5Å	<4Å
1sn3	208	14	-	-	608	29	-	-
1r69	1429	337	28	4	6338	1512	203	19
1ctf	2641	426	39	2	-	-	-	-
1ubq	2038	143	7	-	-	-	-	-
2cro	2705	584	75	5	15040	3873	637	58
3icb	4626	637	32	-	-	-	-	-
4pti	507	48	4	-	6333	675	30	-
4rxn	109	7	-	-	1267	122	8	-
	Simons et. al. [SKHB97]							
1r69	27	21	8	1				
1ctf	16	6	-	-				
2cro	39	18	8	-				
4icb	31	17	2	-				

Table 4: Above: The number of structures, under a certain threshold, produced with 100 and 1 000 saved structures every round in the exhaustive search step using method rmsd. Below: The structures produced by Simons et. al. [SKHB97]. The protein 4icb has 99 percent sequence identity with 3icb.

4 Results

4.1 Amount of structures produced

To study the total number of structures that passed our compactness and overlap test, we tried to apply it to known distributions. For this purpose, we plotted the fraction of structures with a certain rmsd (logarithmic scale) against rmsd in figure 5.

In this study, the distribution of structures with respect to rmsd-values resembles gaussian distributions. A χ^2 test showed that the probability of the region 4.25<rmsd<9.75 of 1r69 with method rmsd being a N(11,2) distribution was 0.01. This is not a very high probability, but considering that the rmsd-values are not uniformly distributed (only non-overlapping structures allowed) and thus the distribution cannot be considered randomly distributed, (which is the case a gaussian distribution describes) this is quite a high correlation. For the structures with rmsd>10 the distribution is distorted by the selection process and there are not as many structures with high rmsd as expected by a gaussian distribution.

The number of produced structures for the different methods differed significantly. The psiblast method produced about twice as many structures as the rmsd method, which produced about ten times as many structures as the random method. The reason for this is that the variation between the fragments within one subset is greater.

The more structures that were produced in total, the better structures were acquired. When saving 1 000 structures instead of 100 in each cycle in the exhaustive search step (see section 3.4) the quality of the best structures was improved. The fraction of good structures was however unchanged, see table 4.

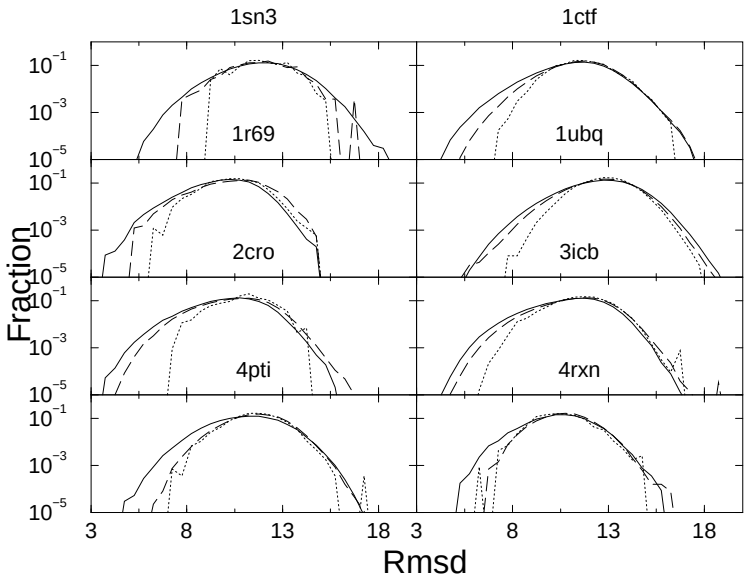

Figure 5: The fraction of structures within a certain rmsd value. The solid line represents the rmsd-method structures, the dashed line represents the psiblast-method structures and the dotted line represents the random-method structures.

4.2 Quality of structures produced

What is clear from the results is that the quality of the fragment does matter. The rmsd method produced a higher fraction of good structures than the other methods in all cases, and the psiblast method produced more good structures than the random method in all alpha proteins. In table 5 the fraction of structures with an rmsd of less than 7 Ångström are shown.

As expected, the two homologs 1r69 and 2cro gave very similar results.

In the psiblast method, the two homologs 1r69 and 2cro were run with the distant homolog 1adr (not the simulation shown in figure 5) among the proteins from which fragments were taken. This rendered fragments of higher

Protein	Good	PsiBlast	Random
1ctf	1.18	0.28	0.02
1r69	5.60	3.79	1.10
1sn3	0.29	0.00	0.00
1ubq	0.10	0.05	0.00
2cro	5.12	3.39	0.09
3icb	2.13	1.03	0.12
4pti	0.95	0.09	0.07
4rxn	2.10	0.21	0.52
ALL	2.18	1.10	0.24

Table 5: Fraction (in %) of the final models that have an rmsd of less than 7Å

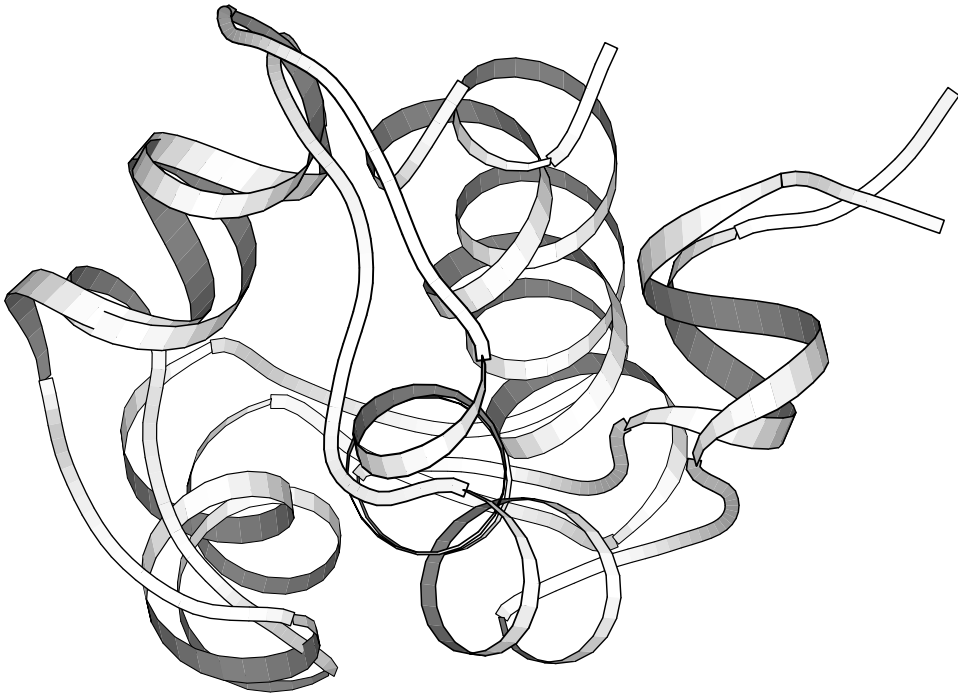

Figure 6: Example of structure, with rmsd 3.8 Å, produced by the rmsd method. This was the best structure acquired for this protein. The structure is superimposed on the native structure: λ-repressor with Brookhaven code 1r69. The figure was made using the program MOLSCRIPT [Kra91].

quality, and increased the performance almost up to the performance of the rmsd method.

The protein 1ubq method psiblast was run with a close homolog among the proteins from which fragments were taken. This means that this simulation gave much better results than expected (see figure 5). In fact it did almost exactly as good as the rmsd method for this protein.

An interesting observation though, is that for all three proteins where homologs were used (1ubq, 1r69 and 2cro) the rmsd method, where no homologs were used in the selection process, still performed better. This indicates that the procedure needs several good fragments at each position in order to work.

In the β-proteins the psiblast method did not do any better than the random method, which can be explained with that the fragment quality was not significantly better. As a reference, fragments 4rxn was run with a method that was equal to the rmsd method in everything but the fact that more proteins to pick fragments from were used (data not shown). This increased the fragment quality significantly. It also increased the fraction of good structures a little, but the performance was not as good as for the α-proteins. Structures with low rmsd-values can be obtained simply by running the simulation in more steps, but these still lack proper β-sheets.

Unlike many other sampling methods for proteins, this is not a lattice model and hence has the possibility to obtain better secondary structures. This method gets better secondary structures provided that the prediction of the fragments is good enough. The criterion of maximum compactness is working quite well, because globular proteins are very compact. If there is one long terminal helix however, there is a tendency to break the helix into two helices and thus avoid that the helix points away from the rest of the protein, reducing the compactness.

In order to get a reference, the structures were compared with the set of decoys used by Park et. al. These are available at: http://dd.stanford.edu/ddownload.cgi?4state_reduced. This comparison is not entirely fair since this set has a much higher fraction of structures with low rmsd-values; structures with low rmsd-values tend to have a higher degree of correct secondary structure.

The method used in this study gave significantly higher degrees of secondary structure preservation for the alpha proteins than the reference structures with both the rmsd method and the psiblast method. For the $\alpha + \beta$ proteins the performance is correlated with the degree of helices in the protein. The rmsd method is still doing better than the reference structures, while the psiblast method is doing worse. For the protein 4rxn with only β-structure neither of the methods nor the reference structures are significantly better than the random method. The fraction of correct secondary structure assignments is tabulated in table 6.

4.3 Energy function

To be able to get the native-like state out of the ensemble of structures created, we need an energy function that can differentiate between native-like and non-native-like folds. We chose to study an energy function, previously studied by Daniel Öhman at SBC, that had reasonable success on the decoy set used by Park et. al. [PHL97].

The energy function evaluated (equation (16)) was able to pick out the x-ray conformation among the acquired structures with a resolution of about 1 in 1 000 structures. The low scores were relatively evenly distributed with respect to rmsd. The ranks and Q-scores are tabulated in table 7. The Q-score is defined as:

$$Q = -\log \frac{rank}{total} \tag{17}$$

The energy function was not able to differentiate between the structures with low rmsd and the ones with high. In figure 8, the fraction within a certain rmsd value range with a good score of the energy function is tabulated against rmsd. The energy function is plotted against the rmsd in figure 7 for the protein 1ubq.

5 Discussion

In these simulations, it has proven difficult to achieve structures with rmsd-values lower than 3. This confirms previous results of Bowie et al. [BE94],

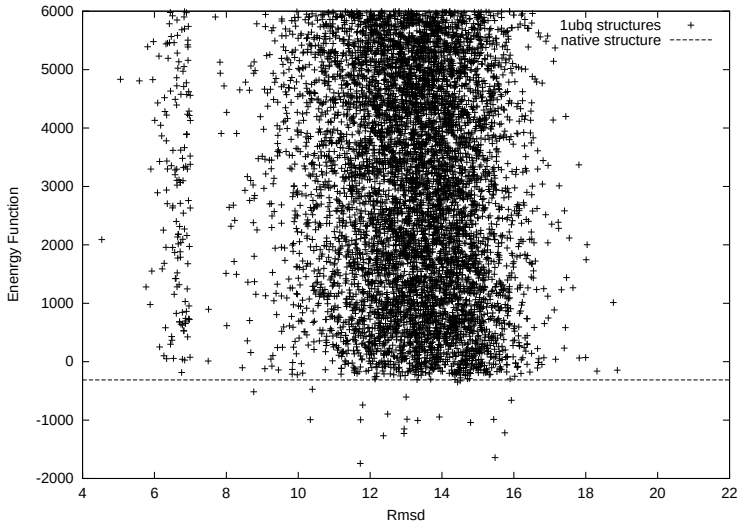

Figure 7: The energy function (equation (16)) is plotted against the rmsd for the protein 1ubq. A border at x= 7Å can be seen, because only every 100:th structure saved over 7 Ångström rmsd is shown.

where a Metropolis simulation using the distance matrix error (dme) as scoring function was used. 100 simulations with 10 000 cycles each were run and the best structures aquired was 2.97 for Cro

In table 4, four simulated proteins in Simons' study are compared with our simulations. In all cases, more and better structures were acquired in this study. This comparison is not fair however, firstly, the fragments used in our simulations are achieved with the rmsd method while Simons uses a statistical method [HB95]. Secondly, the number of saved structures are 1 000-10 000 times greater in our simulations. Third, the number of cycles run in each simulation are 10-100 times greater in our simulations.

The folding of 1r69 has been simulated in several previous studies. Our

Protein	Good structures				All structures			
	PL	Good	PsiBlast	Random	PL	Good	PsiBlast	Random
1ctf	0.586	0.665	0.530	-	0.473	0.649	0.503	0.253
1r69	0.685	0.890	0.755	0.365	0.588	0.874	0.702	0.359
1sn3	0.734	0.691	-	-	0.695	0.746	0.573	0.308
1ubq	-	0.647	0.629	-	-	0.632	0.465	0.439
2cro	0.717	0.889	0.740	-	0.626	0.877	0.684	0.338
3icb	-	0.827	0.736	0.585	-	0.819	0.695	0.504
4pti	0.675	0.687	0.500	-	0.611	0.677	0.527	0.476
4rxn	0.707	0.700	0.636	-	0.673	0.700	0.590	0.625
ALL	0.684	0.749	0.647	0.475	0.611	0.747	0.592	0.413

Table 6: Fraction of correct secondary structure assignments, good structures are defined as structures with rmsd > 7Å.

Protein	Good	PsiBlast	Random	Q	PL	PL Q
1ctf	3/7444	1/2797	1/80	3.53	16/631	1.60
1r69	1/1907	1/1583	1/23	3.55	2/676	2.53
1sn3	108/2248	2/30	1/9	1.32	42/661	1.20
1ubq	17/60894	4/11598	1/798	3.56	-	-
2cro	2/367	5/4411	1/2	3.13	7/675	1.98
3icb	4/9379	12/18621	1/839	3.28	59/654	1.04
4pti	41/1821	18/872	1/31	1.67	67/688	1.01
4rxn	1/224	1/137	2/14	2.27	43/678	1.20
ALL				2.79		1.509

Table 7: ranks and Q-scores for the X-ray structure among the acquired structures. On the right, the ranks and Q-scores for the decoy set used by Park et. al. [PL96], [PHL97].

best structure (not counting the ones where 1 000 structures were saved in each round of the exhaustive search step) had an rmsd of 3.8 Å and is seen in figure 6. Simons et. al. [SKHB97] acquired a model with an rmsd of 3.8 Å and had 8 out of 100 structures with an rmsd of less than 5 Å. Monge et. al. [MLG+95] ran simulations with fixed secondary structure and a genetic algorithm. He had 7.5 percent of the structures under an rmsd of 7 Å, but none under 5 Å. Sun et. al. [STD95] also ran simulations with fixed secondary structure and a genetic algorithm but only acquired structures over 10 Å. For Bowie et. al. [BE94], 33 percent of 200 folding trials yielded structures with a distance matrix error (dme) of less than 4 Å. They were however using knowledge about the protein in the scoring function.

For the hard targets, the psiblast method, did not do significantly better than random. The rmsd method was significantly better than random in all cases, but still performed poorly.

The main downside of this method seems to be the lack of proper beta-sheets. This is partly due to difficulties for methods like psiblast to find fragments with the same fold as the target fold, but the main problem is that the algorithm does not have any reason to form β-sheets. There are some structures having "β-sheet like" structures, which would easily evolve into β-sheets if that would benefit its score.

The choice of fragment length of 9 amino acids is not optimal for β-sheets since the β-strands are usually 4-5 amino acids long [BTB00]. The results for β-proteins could improve by reducing the fragment lengths to 5 residues. This would however greatly increase the number of non-overlapping fragments and thus render an exhaustive search unmanageable.

Although the scoring function tried here was not able to pick out the structures with low rmsd, it could still be useful to remove structures with obvious unreasonable folds. By visual inspection of the structures it seems that many of the high (bad) scoring structures did not look like proteins.

5.1 Why does this work?

The reason that this simple procedure is able to produce native-like structures is that the proteins simulated are very compact in its native state. The criterion

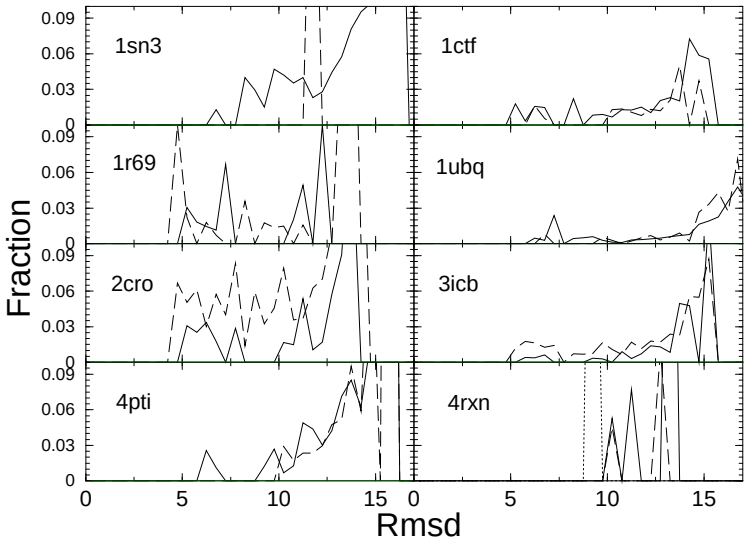

Figure 8: The fraction within a certain rmsd value range which have a good (<200) score. The solid line represents the rmsd-method structures, the dashed line represents the psiblast-method structures and the dotted line represents the random-method structures.

for overlap used in this study is quite strict, and few structures with gyration lower than the native structures passes it.

5.2 What did not work

A method in which the structures were optimised according to the scoring function described under methods was tried. This rendered structures with very low scores that did not look like proteins and were not very compact.

A pure Monte Carlo method were tried. The problem with this approach is that it is hard to find a simulation temperature that optimises the gyration without getting stuck at one particular low-scoring structure.

5.3 Further development

In future development there should be some sort of bonus for beta-sheets in the Monte Carlo step. This could probably be accomplished by giving a bonus to hydrogen bond formation between residues with a distance of say >4. Care has to be taken though so that compactness is still obtained.

The obvious thing that has to be done to make this method useful is of course a scoring function that is able to pick out the native-like folds. This scoring function should use criteria others than the criteria used to evolve the structures.

6 Acknowledgements

I would like to thank my supervisor Dr Arne Elofsson for support on everything. I thank Dr David Liberles for help of calculating sequence similarities. I would also like to thank everyone working at SBC for their help and feedback.

References

[AG96] S. F. Altschul and W. Gish. Local alignment statistics. *Methods in Enzymology*, 266:460–480, 1996.

[AGM+90] S. F. Altschul, W. Gish, W. Miller, E.W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[AK98] S.F. Altschul and E.V. Koonin. Iterated profile searches with psi-blast - a tool for discovery in protein databases. *Trends Biochem. Sci.*, 23:444–447, 1998.

[AMS+97] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[BE94] J. U. Bowie and D. Eisenberg. An evolutionary approach to folding small α-helical proteins that uses sequence information and an empirical guiding function. *Proc. Natl. Acad. Sci.*, 91:4436–4440, 1994.

[BLE91] J. U. Bowie, R. Lüthy, and D. Eisenberg. A method to identify protein sequence that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.

[BTB00] C. Bystroff, V. Thorsson, and D. Baker. Hmmstr: a hidden markov model for local sequence-structure correlations in proteins. *Journal of Molecular Biology*, 301:173–190, 2000.

[CAS00] CASP. The casp www-site. http://predictioncenter.llnl.gov/casp4, 2000.

[Edh90] O. Edholm. *Dator simuleringar i statisk fysik.* 1990.

[ELGD95] A. Elofsson, S. M. Le Grand, and Eisenberg D. Local moves: An efficient algorithm for simulation of protein folding. *Proteins: Structure, function, and genetics*, 23:73–82, 1995.

[Elo93] A. Elofsson. *Molecular Dynamics Simulations as a Tool to Describe Thermodynamical Properties of Proteins.* Karolinska Institutet, 1993.

[Elo00a] A. Elofsson. Dynamic programming. http://www.sbc.su.se/~arne/kurser/swell/dynprog.htm, 2000.

[Elo00b] A. Elofsson. Introduction to sequence alignments. http://www.sbc.su.se/~arne/kurser/swell/alignments_intro.html, 2000.

[Elo00c] A. Elofsson. Pairwise alignments. http://www.sbc.su.se/~arne/kurser/swell/pairwise_alignments.htm, 2000.

[FA95] D. Frishman and P. Argos. Knowledge-based protein secondary structure assignment. *Proteins: Struct. Funct. Genet.*, 23:566–579, 1995.

[FSBM94] K. Fidelis, P.S. Stern, D. Bacon, and J Moult. Comparison of systematic search and database methods for constructing segments of protein structure. *Protein Eng.*, 7:953–960, 1994.

[GBSB00] T. Gilliland, T. Bhat, I. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[HB95] K. Han and D. Baker. Recurring local motifs in proteins. *Journal of Molecular Biology*, 251:176–187, 1995.

[JTT92] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new appoach to protein fold recognition. *Nature*, 358:86–89, 1992.

[KA90] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci.*, 87:2264–2268, 1990.

[Kra91] P. Kraulis. Molscript: a program to produce both detailed and schematic plots of protein structures. *J. Appl. Crystallogr*, 24:946–950, 1991.

[KS96] A. Kolinski and J. Skolnick. *Lattice models of protein folding, dynamics and thermodynamics.* R.G. Landes Austin Texas, 1996.

[Lev68] C. Levinthal. Are there pathways for protein folding? *J. Chem. Phys*, 65:44–45, 1968.

[MFH94] A. Monge, R. A. Friesner, and B. Honig. An algorithm to generate low-resolution protein tertiary structures from knowledge of secondary structure. *Proc. Natl. Acad. Sci.*, 91:5027–5029, 1994.

[MHB+97] J. Moult, T. Hubbard, S. H. Bryant, K. Fidelis, and J. T. Pedersen. Critical assessment of methods of proteins structure predictions (CASP): Round II. *Proteins: Struct. Funct. Genet., Suppl*, 1:2–6, 1997.

[MLG+95] A. Monge, E. J. P. Lathrop, J. R. Gunn, P. S. Shenkin, and R. A. Friesner. Computer modelling of protein folding: conformational and energetic analysis of reduced and detailed protein models. *Journal of Molecular Biology*, 247:5027–5029, 1995.

[MRRT53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[OKR+99] A. R. Ortiz, A. Kolinski, P. Rotkiewicz, B. Ilkowski, and J. Skolnick. Ab initio folding of proteins using restraints derived from evolutionary information. *Proteins: Structure, function, and genetics*, 3:177–185, 1999.

[PHL97] B. H. Park, E. S. Huang, and M. Levitt. Factors affectin the ability of energy functions to discriminate correct from incorrect folds. *Journal of Molecular Biology*, 266:831–846, 1997.

[PL96] B. H. Park and M. Levitt. Energy functions that discriminate x-ray and near-native folds from well-constructed decoys. *Journal of Molecular Biology*, 258:367–392, 1996.

[RS93] B. Rost and C. Sander. Prediction of protein secondary structure structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.

[Sal93] A. Sali. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234(12):779–815, 1993.

[SKHB97] K.T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of Molecular Biology*, 268:209–225, 1997.

[STD95] S. Sun, P. T. Thomas, and K. A. Dill. A simple protein folding algorithm using a binary code and secondary structure constraints. *Protein Eng.*, 8:769–778, 1995.

[Str95] Lubert Stryer. *Biochemistry fourth edition.* W. H. Freeman and Company New York, 1995.

[SXHL99] R Samudrala, Y. Xia, E. Huang, and M. Levitt. Ab initio protein structure prediction using a combined hierarchical approach. *Proteins: Structure, function, and genetics*, 3:194–198, 1999.

A Parameters of the scoring function.

Table 8: Contact Distances: Daniel Öhman, kT units

	C_α	ALA	GLY	VAL	LEU	ILE	PHE	TYR	TRP	MET	PRO	SER	THR	CYS	ASN	GLN	ASP	GLU	LYS	ARG	HIS
HIS	5.9	6.0	6.0	6.2	6.0	6.1	6.1	6.0	5.9	6.0	5.9	5.9	6.0	5.9	5.9	6.0	5.8	6.0	6.1	6.1	6.1
ARG	5.9	6.1	6.2	6.3	6.2	6.2	6.2	6.1	6.2	6.1	6.1	6.0	6.1	6.0	6.0	6.1	5.8	6.0	6.2	6.3	
LYS	5.9	6.1	6.1	6.3	6.2	6.3	6.3	6.2	6.1	6.2	6.1	6.0	6.0	6.2	5.9	6.0	5.8	5.9	6.2		
GLU	6.0	6.0	6.1	6.2	6.1	6.2	6.2	6.1	6.1	6.2	6.0	5.8	5.9	6.1	5.9	6.1	5.9	6.2			
ASP	5.9	5.8	5.9	6.1	6.1	6.2	6.0	6.0	6.0	6.0	5.9	5.6	5.7	6.0	5.7	6.0	5.6				
GLN	5.9	6.1	6.3	6.2	6.1	6.2	6.2	6.1	6.1	6.1	6.1	5.9	6.0	6.2	5.9	6.0					
ASN	5.8	6.0	5.9	6.1	6.1	6.2	6.2	6.0	6.0	6.2	6.1	5.9	5.9	6.1	5.8						
CYS	5.8	5.8	6.0	5.9	5.9	5.9	5.9	5.9	6.0	5.8	5.9	5.8	6.2	5.3							
THR	5.9	6.0	5.9	6.3	6.2	6.2	6.2	6.1	6.2	6.1	6.1	5.8	6.0								
SER	5.8	5.9	5.9	6.1	6.1	6.2	6.1	6.0	5.9	6.0	5.8	5.8									
PRO	6.1	5.8	5.8	6.1	6.0	6.2	6.1	6.1	5.9	6.0	6.0										
MET	5.9	5.9	6.2	6.1	5.9	6.0	5.9	6.0	6.0	5.9											
TRP	5.9	6.0	6.2	6.0	6.0	6.0	6.0	6.0	6.0												
TYR	5.9	6.0	6.0	6.1	6.0	6.0	6.0	6.0													
PHE	5.9	6.0	6.1	6.0	5.9	6.0	5.8														
ILE	5.9	5.9	6.2	6.0	5.9	5.9															
LEU	5.9	5.9	6.3	5.9	5.9																
VAL	5.9	6.0	6.2	6.0																	
GLY	5.8	6.0	5.6																		
ALA	5.8	5.9																			
C_α	5.7																				

Table 9: Contact Energies: Daniel Öhman, kT units

	C_α	ALA	GLY	VAL	LEU	ILE	PHE	TYR	TRP	MET	PRO	SER	THR	CYS	ASN	GLN	ASP	GLU	LYS	ARG	HIS
HIS	-0.7	-0.7	-0.6	-0.8	-0.8	-0.8	-0.9	-0.9	-0.9	-1.0	0.1	-0.6	-0.6	-1.0	-0.4	-0.6	-0.5	-0.6	-0.4	-0.6	-1.0
ARG	-0.7	-0.9	-0.5	-0.7	-0.9	-0.8	-0.7	-0.8	-0.8	-0.9	-0.1	-0.6	-0.6	-0.6	-0.6	-0.9	-0.9	-1.1	-0.5	-0.7	
LYS	-0.6	-0.7	-0.4	-0.7	-0.7	-0.7	-0.6	-0.7	-0.6	-0.6	0.0	-0.5	-0.5	-0.5	-0.6	-0.8	-0.9	-1.1	-0.6		
GLU	-0.6	-0.8	-0.3	-0.5	-0.6	-0.6	-0.5	-0.5	-0.5	-0.6	0.2	-0.5	-0.5	-0.4	-0.5	-0.8	-0.4	-0.7			
ASP	-0.5	-0.7	-0.4	-0.4	-0.5	-0.4	-0.4	-0.4	-0.4	-0.5	0.3	-0.5	-0.5	-0.4	-0.6	-0.7	-0.3				
GLN	-0.7	-0.8	-0.5	-0.7	-0.8	-0.7	-0.7	-0.8	-0.8	-0.8	-0.1	-0.7	-0.7	-0.5	-0.7	-1.0					
ASN	-0.6	-0.6	-0.6	-0.5	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	0.2	-0.6	-0.6	-0.6	-0.7						
CYS	-0.8	-0.8	-0.8	-0.8	-0.9	-0.9	-0.9	-0.9	-0.8	-0.9	-0.1	-0.7	-0.7	-2.2							
THR	-0.6	-0.6	-0.6	-0.8	-0.8	-0.8	-0.7	-0.7	-0.8	-0.7	0.1	-0.6	-0.7								
SER	-0.6	-0.7	-0.6	-0.6	-0.6	-0.7	-0.7	-0.6	-0.7	-0.6	0.1	-0.6									
PRO	-0.0	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1	-0.0	-0.1	-0.1	0.5										
MET	-0.8	-0.9	-0.7	-1.1	-1.1	-1.1	-1.0	-1.0	-0.9	-1.2											
TRP	-0.8	-0.8	-0.8	-0.9	-0.9	-1.0	-1.0	-1.0	-1.2												
TYR	-0.8	-0.9	-0.7	-1.0	-1.0	-1.1	-1.0	-1.0													
PHE	-0.8	-0.8	-0.7	-1.0	-1.1	-1.1	-1.1														
ILE	-0.9	-0.9	-0.7	-1.2	-1.2	-1.3															
LEU	-0.9	-1.0	-0.7	-1.1	-1.3																
VAL	-0.8	-0.9	-0.6	-1.2																	
GLY	-0.6	-0.8	-0.9																		
ALA	-0.8	-1.2																			
C_α	0.0																				