**Proceedings of the Fourth IASTED International Conference**
**Advances in Computer Science and Technology (ACST 2008)**
**April 2-4, 2008 Langkawi, Malaysia**
**ISBN CD: 978-0-88986-730-7**

# OPTIMAL LENGTH OF FRAGMENTS FOR USE IN PROTEIN STRUCTURE PREDICTION

Trent Higgs Bela Stantic Md Tamjidul Hoque
Institute for Integrated and Intelligent Systems
Griffith University, Queensland, Australia
{T.Higgs, B.Stantic, T.Hoque}@griffith.edu.au

**ABSTRACT**
Proteins are three-dimensional structures that carry out many of the vital biological functions in organisms. Because structure, not amino acid sequence order, carry out certain functions, it is important to understand how proteins fold. Computational methods for protein structure prediction mentioned in the literature are computationally demanding. To reduce computational demand fragment libraries were introduced. Fragment libraries work by taking short segments of the polypeptide chain and limiting the amount of conformations that will be considered for a particular segment. In this paper an extensive analysis towards finding the optimal length of fragments contained within fragment libraries was conducted. An extensive analysis was done on protein structures stored in a ORDBMS to exploit its power. Experiments focused on the structural similarity between fragments of identical primary protein sub-sequence within different proteins, and amount of occurrences of similar or closely similar fragments within different proteins. Experimental results indicate that short to medium sized fragments have stronger structural correlations with matching fragments within different proteins.

**KEY WORDS**
Bioinformatics, Fragment Libraries, and Protein Structure Prediction.

## 1 Introduction

Proteins are three-dimensional structures. They carry out many of the vital biological functions in organisms. Due to this, there is a need to understand how proteins fold into their final three-dimensional structures and interact with one another. Computational methods that have been developed to solve this problem are comparative modelling, threading, and ab initio.

Comparative modelling and threading can produce inaccurate models, and ab initio is very computationally demanding. To solve this problem fragment libraries were developed.

The idea of being able to successfully predict a proteins three-dimensional structure is a mystery that has baffled scientists for many years. The reason why a solution to the protein folding problem has been heavily sought after is due to their importance. Proteins carry out all of the main functionality within an organism on a cellular level. For example, red blood cells contain a protein known as the hemoglobin. This protein carries out the functionality of carrying oxygen to the blood stream.

A byproduct of proteins carrying out vital biological functions is that if certain proteins do not fold correctly then numerous diseases may present themselves. These include: Alzheimers Disease, Cystic Fibrosis, and numerous prion diseases [5]. By knowing how a protein folds into its unique three-dimensional structure we would have a better understanding of why these proteins are not folding properly, and hence would be able to do something about it. Another benefit of being able to predict the three-dimensional structures of proteins is the ability to use that technology to design proteins that will carry out a specific biological task [4]. By doing this scientists will have the ability to cure or prevent human diseases/ilnesses by using these specifically designed proteins.

To improve the quality of protein structure prediction numerous computational methods have been proposed. Computational protein structure prediction methods can be grouped into the following categories: comparative modelling, threading, and ab initio.

Comparative modelling and threading both have problems if the sequence similarity is low. However, ab initio predicts a structure from the proteins sequence alone, hence it does not require template sequence information like the other two methods. The main limitation of the ab initio approach is the high computational cost. To minimise this problem fragment libraries have been introduced, which are used to limit the amount of conformations considered for a particular segment of the protein chain.

In this paper we have conducted an extensive analysis of the structural similarity between various sized matching fragments within different proteins, to elicit the optimal fragment size for use within fragment based protein prediction software. This was to investigate how fragment length could improve/decrease the accuracy of the protein structure prediction process.

The reminder of this paper is organized as follows: in Section 2 we present background information, Section 3 discusses the methodology we implemented, Section 4 presents and discusses our experimental results, and in Section 5 we conclude our findings and mention possible future work.

## 2 Background

A protein is made up of a collection of amino acids, which are molecules that have both carboxyl and amino groups. An amino acid contains a carbon atom (C$\alpha$), and has four different connections, these include an amino group, carboxyl group, a hydrogen atom, and a side chain (this differs depending on the amino acid). The C$\alpha$ atom is the central atom of the amino acid and all of the other connectors are attached to it. The C$\alpha$ atom is important as it is the central atom in every amino acid and has a great deal of impact on the backbone conformation of a protein [6].

Proteins can take on an enumerable amount of conformations. Even for the simplified assumption [1] that if each amino acid can have 3 degrees of rotation, a protein chain that has 200 residues could at the very minimum have $3^{200}$ possible conformations, which is an astronomical number. Hence, it is very hard to predict a proteins three-dimensional structure by searching all possible structures available. To alleviate this problem 3 main computational methods to predict a proteins structure have been developed, these are: comparative modelling, threading, and ab initio.

Comparative modelling and threading work by aligning a protein's target sequence with one or more template sequences. Ab initio on the other hand is based off of the principle that a proteins native structure is at its lowest free energy minimum [2]. This means instead of using template sequences, it predicts a proteins structure based off of its sequence alone by searching the free energy conformational space. The major flaw with this methodology is that the conformational space is too large. To solve the heavy computational problem fragment libraries were introduced.

Fragment libraries are utilised within protein structure prediction to limit the amount of possible conformations considered for a given protein. They are either: local structure motifs or three-dimensional structure motifs [11].

Local structure motifs refer to one or more secondary structures joined together, and three-dimensional structure motifs are fully conformed protein structures [11]. Secondary structures can be either a alpha helix, or a pleated beta sheet, and are structures created by hydrogen bonding when the protein chain begins to fold. Therefore, it can be seen that a fully conformed protein is made up of numerous local sequence motifs. The state-of-the-art protein prediction software currently available at the moment mainly use local structure motifs for their predictions, because using fully conformed proteins would drastically increase the CPU time.

Fragment based protein structure prediction software, like Rosetta [7] [8] and Tasser [9] [10] do not allow for their fragment length to be changed, and therefore there is no way of proving the quality of the fragment being used. It can be perceived that increasing/decreasing the fragment length could greatly improve the accuracy of the prediction process, which is still far off being perfect. There also has been no real investigation into the structural similarity between identical fragments within different proteins to determine what the optimal fragment length could be. Instead, the research in structural similarity has been heavily focused on devising ways to quickly match sub-structures of any size within different proteins [6] [11], rather then focusing on the fragment length.

## 3 Fragment Similarity Prediction

To find the optimal length of fragments to be used within fragment based protein prediction software we have carried out an investigation into protein fragment similarity. This incorporated taking already folded proteins from the PDB (Protein Data Bank) and checking how often particular motifs/fragments of a set size within each protein in the protein database matched with every other protein sequence contained within the database.

For example, if one of the motifs contained the amino acids alanine, threonine, and glycine (ATG) right after one another. ATG would be searched for throughout every protein sequence in the database and every occurrence of it that appeared in the exact same order (i.e. A as the first amino acid in the motif, T as the second amino acid in the motif, and G as the third amino acid in the motif) would be recorded and analysed. Other than recording matches the distance between the amino acids (i.e. between the C$\alpha$ atoms) of the motif/fragment were also stored to determine how structurally similar identical fragments are within different proteins.

To conduct this analysis we took advantage of the powerful ORDBMS (Object Relational Database Management System) engine. All protein data was converted into a suitable format, and stored within a Oracle 10g database. This database contained a pdb table to store all of the protein data, a match table that contained all matching fragments and their Root Mean Squared Distance (RMSD) value, and a report table that contained a summary of all matching fragments, which included the number of occurrences of a particular fragment, and the minimum and maximum RMSD values for that fragment. The RMSD equation is used to determine how structurally similar identical fragments (the same sequence of amino acids) are within different proteins [3]. It does this by measuring the distances between the C$\alpha$ atoms of each amino acid within the fragment (see equation 1). We used the root mean squared distance equation rather than other statistical structural measures (e.g. root mean squared deviation) due to it being less costly to calculate, and due to it producing close-enough structural similarity values.

$$rmsd = \sqrt{\sum d_i^2 \div (n(n-1) \div 2)} \quad (1)$$

### 3.1 Fragment Similarity Algorithm

The algorithm that we used for the protein fragment search can be found in Algorithm 1. This algorithm works by first grabbing all of the proteins within the database (protAll). It will then iterate throughout all of protAll so that all proteins within the database are searched. The main body introduces two fragments A and B. A is a protein fragment of size k (where k $\geq$ 4) that starts from the amino acid of a particular protein that protAll is currently on (curr) and ends k-1 amino acids past curr (i.e. A = protAll.currentAcidID to protAll.currentAcidID + k-1). B on the other hand holds many different fragments depending on the first amino acid in A.

**Algorithm 1** Fragment Similarity Algorithm

```
protAll = get all proteins in the database;
while protAll NOT NULL do
  if protAll.prot_name != previous.protAll.prot_name
  then
    Mark previous.protAll.prot_name AS DONE;
  end if
  A() = fragment of protein from proteinAll.acidID to
  proteinAll.acidID + k - 1;
  dist = RMSD for A(1) to A(3);
  Add A(1) ... A(3) dist to match, report;
  protAcid = All amino acids within all proteins that
  contain A(1).acid_name;
  while protAcid is NOT NULL do
    B() = fragment of protein from protAcid.acidID to
    protAcid.acidID + k - 1;
    if A(2).acid_name = B(2).acid_name then
      if A(3).acid_name = B(3).acid_name then
        dist = RMSD for B(1) to B(3);
        Add B(1) ... B(3) & dist into match, report;
        if A(4).acid_name = B(4).acid_name then
          dist = RMSD for A(1) to A(4);
          Add A(1) ... A(4) & dist into match, report;
          dist = RMSD for B(1) to B(4);
          Add B(1) ... B(4) & dist into match, report;
        end if
        Keep checking up to A(k) to B(k);
      end if
    end if
  end while
end while
Mark last protein AS DONE;
```

B is assigned by finding every protein and the related amino acid positions in the database that have the same amino acid as the first acid in A (protAcid). protAcid is then iterated through and each time B is assigned the fragment, k in length (where k $\geq$ 4), generated by protAcids current amino acid for a particular protein (curr) up to k-1 amino acids past curr (i.e. B = protAcid.currentAcidID to protAcid.currentAcidID + k-1). After A and B are both found it is then a simple matter of checking if the amino acid of A at position two matches B's amino acid at position two and then same for position three. If this is the case then root mean squared distance for B is calculated and recorded in database. The same is done for 4, 5, 6, 7...k fragments in length, but there is one slight difference. Due to the algorithm automatically adding A 3 sized fragments to the database as a default, there is a need to calculate the root mean squared distance for A 4, 5, 6, 7...k sized fragment matches and add them to the match and report tables before calculating the RMSD value for B.

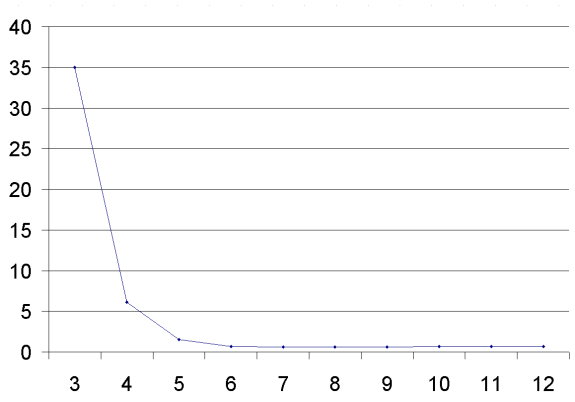


Figure 1. The average difference between minimum and maximum RMSD for each fragment length

## 4 Experimental Evaluation

The fragment analysis, mentioned in the previous section, was performed on a protein database, which contained protein structures stored within the PDB (approximately 24000 structures). We included proteins from PDB that contained single chain only. There are proteins that have amino acids, which are made up of more then one C$\alpha$ atom, and therefore were not suitable for our testing because RMSD values can not be calculated properly for fragments that have amino acids with more than one C$\alpha$ atom.

All experimental results presented in this section are computed on a Sun Fire V880 server with 8 x UltraSPARC-III 900MHZ CPU using 8GB RAM, running Oracle 10g RDBMS. The Database block size was 8K, SGA (System Global Area) size was 1GB, and fragment lengths 3-12 were considered.

Our goal within this experimentation stage was to find out what the most appropriate fragment length should be within fragment based protein predictions software (e.g. Rosetta). To achieve this after the results for the fragment similarity analysis were produced, we ran a second experiment to determine the optimal fragment size/s to be used in Rosetta. This was done by changing the fragment size within Rosetta in place of its already existing 9 sized local-sequence motifs that it uses for backbone protein predictions. Each fragment length that was used within the fragment similarity analysis was tested with this modified

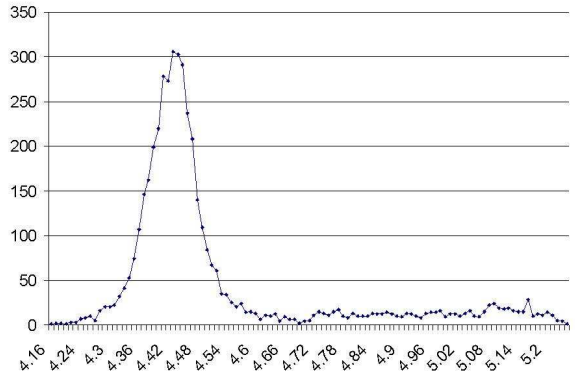


Figure 2. Number of occurrences of RMSD for the max occurring fragment of length 3

Rosetta. By doing this we are able to see which one of these lengths produce more accurate predictions, so that we can determine the optimal fragment size. In order to determine optimal fragment length we modified Rosetta's fragment generation functionality to allow different sized fragments to be generated and we also modified the program itself so that different sized fragments could be used for structure prediction.

### 4.1 Results

In Figure 1 are shown the average difference between the minimum and maximum root mean squared distance (RMSD) values for a particular fragment length (the x axis is the fragment size, and the y axis is the RMSD value). In Table 1 we present the maximum occurring fragments for a set length (i.e 3-12). The fragment name in Table 1 refers to the amino acids that make up that fragment, these are written in the three-letter notation. For example, ALAALAALA refers to a fragment composed of the amino acids alanine, alanine, and alanine. In Figures 2, 3, 4, 5 and 6 are results for the number of occurrences of RMSD values for these max occurring fragments for sizes 3-12. The x axis shows the RMSD value, and the y axis shows the number of occurrences for that RMSD value. In table 2 is the analysed results from these graphs, it shows the percentage of overall data that falls within the bell curve (i.e. occurrences of identical fragments within different proteins), and the differences between the maximum and minimum ranges of that bell curve.

In Table 3 are the results of running the different fragment sizes through the modified version of Rosetta. It contains the CPU time (hh:mm:ss) required to complete 100 decoys (note that everything was rounded up to minutes) for the BAX protein (1f16) for a particular fragment size, and the maximum and minimum scores for each fragment size.

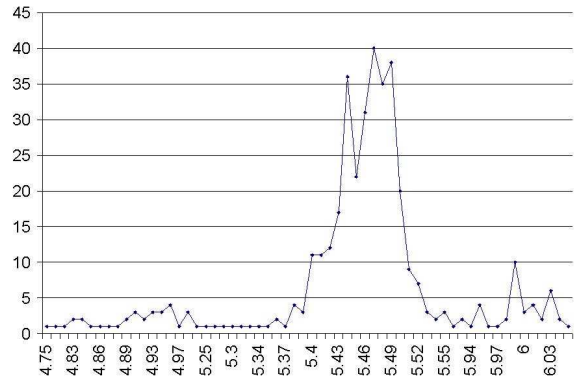


Figure 3. Number of occurrences of RMSD for the max occurring fragment of length 5

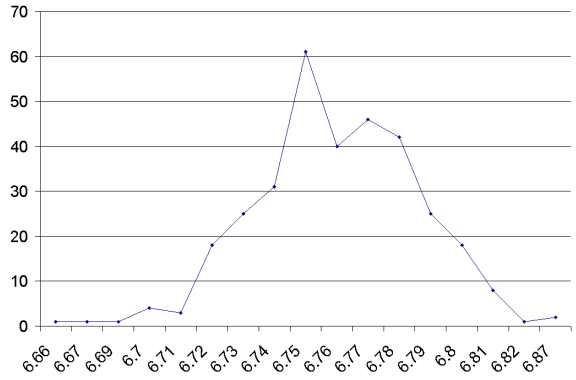


Figure 4. Number of occurrences of RMSD for the max occurring fragment of length 9

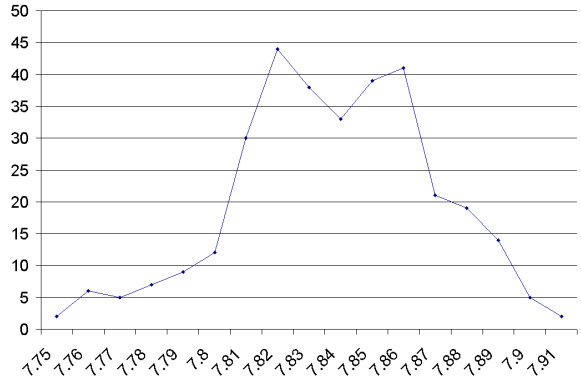


Figure 5. Number of occurrences of RMSD for the max occurring fragment of length 11

| Fragment Size | Fragment Name | Occurrences | Min RMSD | Max RMSD |
|---|---|---|---|---|
| 3 | ALAALAALA | 4452 | 4.161 | 7.328 |
| 4 | GLYSERSERGLY | 657 | 4.2 | 6.692 |
| 5 | LEUASPALAVALARG | 393 | 4.748 | 6.692 |
| 6 | VALASPALAALAVALARG | 334 | 5.148 | 5.575 |
| 7 | GLYILEGLYHISLEULEUTHR | 331 | 5.764 | 9.896 |
| 8 | GLYILEGLYHISLEULEUTHRLYS | 330 | 9.035 | 10.901 |
| 9 | ASPGLUALAGLULYSLEUPHEASNGLN | 329 | 6.544 | 6.872 |
| 10 | ASPGLUALAGLULYSLEUPHEASNGLNASP | 329 | 7.07 | 7.583 |
| 11 | ASPGLUALAGLULYSLEUPHEASNGLNASPVAL | 329 | 7.609 | 7.911 |
| 12 | LYSASPGLUALAGLULYSLEUPHEASNGLNASPVAL | 328 | 8.162 | 8.839 |

Table 1. Max occurring fragments used

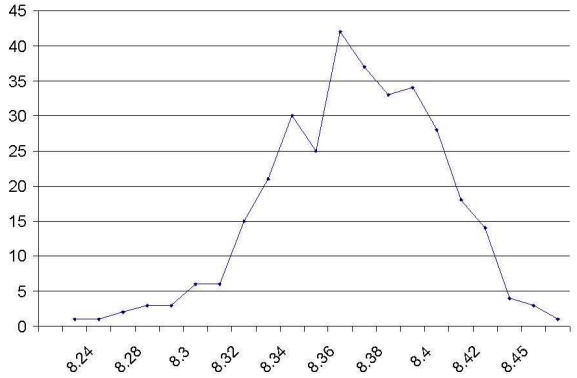


Figure 6. Number of occurrences of RMSD for the max occurring fragment of length 12

| Frag.Size | Frag. within Bell Curve | RMSD diff. |
|---|---|---|
| 3 | 82.21% | 0.34 |
| 4 | 88.49% | 0.18 |
| 5 | 78.86% | 0.53 |
| 6 | 94.91% | 0.15 |
| 7 | 95.77% | 0.19 |
| 8 | 96.06% | 0.20 |
| 9 | 98.77% | 0.13 |
| 10 | 97.87% | 0.14 |
| 11 | 100% | 0.16 |
| 12 | 99.69% | 0.16 |

Table 2. Percentage of data held within bell curve

| Fragment Size | CPU Time | Max Score | Min Score |
|---|---|---|---|
| 3 | 5:23:00 | 26.63 | -98.4 |
| 4 | 5:46:00 | 14.04 | -104.07 |
| 5 | 5:50:00 | -2.26 | -101.67 |
| 6 | 6:13:00 | 0.97 | -98.27 |
| 7 | 6:13:00 | -8.35 | -97.01 |
| 8 | 6:25:00 | -3.25 | -111.68 |
| 9 | 6:34:00 | -28.47 | -116.04 |
| 10 | 6:38:00 | -27.8 | -106.49 |
| 11 | 6:38:00 | -16.19 | -108.17 |
| 12 | 6:50:00 | -25.15 | -105.77 |

Table 3. Accuracy verses CPU Time for different fragment sizes within Rosetta

### 4.2 Analysis and Discussion

In our results we looked at the number of occurrences of root mean squared distance (RMSD) for the maximum occurring fragment for each fragment length (see table 1 to see which fragments we used for the maximum occurring fragment for a particular length, and see figures 2, 3, 4, 5, and 6). This sort of data shows us the spread of data between the max occurring RMSD value (the maximum point in the graph) for each maximum occurring fragment for a particular length. The higher the percentage of data that is contained within the bell curve (the data points that fall before and after the maximum RMSD point) demonstrates that the majority of occurrences of RMSD for a particular fragment length is appearing within a set RMSD range. The smaller the difference between the minimum and maximum values of this RMSD range is, the more structurally similar identical fragments within different proteins are (i.e. for that maximum occurring fragment k in length).

In regards to the number of occurrences of RMSD for the maximum occurring fragment, the obvious conclusion to make is that sizes 3, 4 and 5 (compared with all other fragment sizes) all have, on average, lower structural similarity with identical fragments within different proteins and less occurrences held within a set RMSD range (i.e. bell

curve). This is further proven by Figure 1. This figure depicts the average difference between minimum and maximum RMSD values, for all matched fragments within different proteins for a particular length. This means it gives us a good indication of how structurally similar identical fragments (i.e. matched fragments) of a certain length are within different proteins.

In concerns to accuracy it seems that the longer the fragment ($> 8$ residues) the lower the Rosetta score range is (see table 3). For our purposes here we have decided to base our analysis for accuracy on the overall score Rosetta gives to each decoy it generates. This is the score given by the energy function that Rosetta uses, and the lower the score is the closer it is considered to be to its native structure. As you can see in table 3, fragments from length 8 and up have smaller maximum scores and very low minimum scores. This means that they generate more structures that have low scores and thus are closer to their native structures. Therefore, from this we can conclude that short-medium fragment lengths used for the main backbone predictions, in Rosetta, produce more accurate results then smaller sized fragments.

Fragment sizes 3, 4, and 5 were not suitable, even though more matches of these fragment lengths were found within different proteins. This was due to them all having high structural differences from identical fragments within different proteins. Fragment sizes 6, 7 and 8 all had reasonable structural difference tolerance, but still, like the other fragment lengths, had too high of a score range within Rosetta. That left us with sizes 9, 10, 11 and 12, which all had reasonable structural difference tolerance (on average and within their max occurring fragment). Therefore, we concluded that short-medium sized fragments are more optimal for fragment based software due to them being more structurally similar to matching fragments within other proteins. Overall we found that fragment size 9 performed better then other fragments within Rosetta (lowest score range), and is concluded to be the optimal fragment size for use within Rosetta.

## 5 Conclusion and Future Work

In this work in extensive experimental study we addressed a question of finding the optimal size of fragments to be used within fragment based protein structure prediction software. The most significant results we discovered was that fragment sizes 6-12 were more structurally similar than compared with smaller fragment sizes (e.g. 3-5). And this lead to us finding out that sizes 9-12 sized (short-medium) fragments produced lower score ranges within Rosetta, and the optimal size for Rosetta (out of 3-12 fragment lengths) being 9. This computationally shows that the longer fragment sizes produce more accurate results due to identical fragments within different proteins being more structurally similar.

As for future work it would be interesting to investigate how different biological forces impact on matching fragments structural similarity within different proteins, and use findings to modify the fragment generation process.

## Acknowledgement

This research is partly sponsored by ARC (Australian Research Council) grant no DP0557303.

## References

[1] D. Baker. Proteins by design. *The Scientist*, pages 26–32, July 2006.

[2] D. Baker and A. Sali. Protein structure prediction and structural genomics. *SCIENCE*, 294:93–96, Oct. 2001.

[3] O. Carugo. Statistical validation of the rootmeansquaredistance, a measure of protein structural proximity. *Protein Engineering, Design and Selection*, 20(1):3338, 2007.

[4] N. C. et al. *Biology*. Benjamin Cummings, ISBN: 080537146X, 2004.

[5] Feng Ding, Joshua J. LaRocque, and Nikolay V. Dokholyan . Direct Observation of Protein Folding, Aggregation, and a Prion-like Conformational Conversion. *The Journal of Biological Chemistry*, 280(48):40235–40240, 2005.

[6] Z. Huang and X. Zhou. High dimensional indexing for protein structure matching using bowties. In *Procedings of the 3rd Asia-Pacific Bioinformatics Conference*, pages 21–30, 2005.

[7] 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.

[8] S. KT and B. Ruczinski and I. Baker. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins*, pages 171–176, 1999.

[9] Y. Zhang and J. Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *PNAS*, 101(20):7594–7599, May 2004.

[10] Y. Zhang and J. Skolnick. Tertiary structure predictions on a comprehensive benchmark of medium to large size proteins. *Biophysical Journal*, 87:2647–2655, Oct. 2004.

[11] Zi Huang and Xiaofang Zhou and Dawei Song and Peter Bruza. Dimensionality reduction in patch- signature based protein structure matching. In *Procedings of the 17th Australiasian Database Conference - ADC06*, pages 89–97, 2006.