Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Haitham Ahmad Gamal Abdel Naser

1/12/2009

Abstract

"How do proteins fold?" is one of the most challenging questions in molecular biology. Researchers exerted great efforts to solve this problem over the few last decades. However the enormous number of theoretically possible structures and the wide diversity of existing, already known conformations slow down the wheel of scientific contribution. And unfortunately, the longer the protein under study the more difficult the problem becomes.

This study approaches the problem from a statistical point of view. Taking advantage of the similarity among newly discovered structures and already known ones and guided by the fact that the hydrophobicity of a protein is the key factor in its folding process. The approach introduced here tries to prove the existence of a relationship between the hydrophobicity of the protein's constituent amino-acids and its backbone angles at subsequence level. The study attempts to statistically prove the existence of this relationship through building a library of clustered hydrophobicity patterns, i.e., a library of protein subsequences clustered based on their hydrophobicity, then the central angle measurements of these subsequences are extracted. Finally, the best standard continuous probability distribution that describes the central angle measurements of each cluster is sought using a statistical test known as Kolmogrov-Smirnov (KS) test. This process is repeated for subsequences of different lengths typically 3, 5 and 7. The results of the KS-test are used to assess the strength of the relationship subject to these different lengths.

It was found that there is a considerable improvement in KS-test results of clustered subsequences over unclustered ones i.e. the relationship is

more apparent among proteins with similar hydrophobicity pattern, the thing that emphasis the relationship between hydrophobicity patterns and angle measurements. The study showed also that the longer the subsequences used in clustering, the better the fits. In other words, the goodness of fits is directly proportional to the length of the subsequences used.

Protein Folding Pathway Prediction

By

Haitham Ahmad Gamal Abdel Naser

B.Sc. of Computer Science

Faculty of Computers and Informatics

Zagazig University

Supervised by

Prof. Ibrahim Mahmoud El-Henawy

Dean of Faculty of Computers and Informatics

Zagazig University

Dr. Ahmad Hussein Kamal

Computer Science department

Faculty of Computers and Informatics

Cairo University

Dr. Hisham Al-Shishiny

Chief Scientist

And Manager of IBM Center for Advanced Studies

IBM Cairo Technology Development Center

IBM Egypt

A thesis presented to the department of Computer Science, Faculty of Computers and Informatics, Zagazig University

In fulfillment of the thesis requirement for the degree of Master in Computer Science

Zagazig, Egypt,

2009

Dedication

To my parents

To my wife

To my sister

And

To my country

Table of Contents

List of Figures

Chapter 1
Introduction

1.1 Motivation

Discovering the method by which a protein folds is crucial to many medical and generally biological fields. It helps in identifying malformed proteins which in turn helps in avoiding serious diseases that incorrect structures can produce, like mad cow disease. Protein folding is also very important in the design of drugs; Knowledge of the shape of a protein is fundamental in designing enhancing drugs, if it is useful or suppressing drugs, if it is harmful.

Extensive attempts to solve the problem were made but no unique, complete solution has been found yet. Each researcher adds a new technique or a new piece of information towards the ultimate, complete, and undiscovered yet solution.

1.2 Objective

This study assumes and attempts to empirically prove that there is a strong relationship between the hydrophobicity of the residues of a protein and its backbone angles. Although the relationship between hydrophobicity and the folded structure of a protein is intensively studied and widely accepted, this study claims that a strong relationship exists not only between the final conformation and the whole protein, but also between backbone angles and hydrophobicity pattern of its local subsequence.

This approach aims also to build a statistical library that can be used by researchers working with protein folding to predict backbone angles

using information about the hydrophobicity of the local residues surrounding these angles.

1.3 Approach Summary

In this study a four-phased approach is introduced. First of all, Input data are prepared in two phases which are: angle extraction and chopping. A k-means clustering is performed next. The corner stone of this study is that the similarity function of clustering depends on subsequences hydrophobicity. The final stage takes all the clusters and tries to fit them into one of 66 possible standard continuous probability distributions using a Kolmogrov-Smirnov test.

1.4 Thesis Organization

Chapter 2 discusses the protein structure prediction problem and why it is important and complex. An overview of some important studies and researches in the field of protein folding is presented in chapter 3. The same chapter illustrates the relation between the approach introduced here and the prior art. Chapter 4 discusses the method used in this study in great detail, phase by phase. Chapter 5 finalizes the thesis with a discussion of the results. A conclusion is drawn and possible future work is introduced in chapter 6.

Three appendices can be found at the end of the thesis including some necessary topics. These topics are separated in the form of appendices because they are either not so close to the interest of this study or because they are meant to be used as a reference while discussing other parts (to prevent reader distraction).

Chapter 2

Biological Background and Motivation

2.1 Why to study proteins?

The word "protein" originates from the Greek word "proteus" which means "of the first rank". Proteins constitute much of the bulk of living organisms: enzymes, hormones and structural material. Most of the genes in the genetic makeup of an organism are protein-coding genes i.e. they specify instructions for building proteins. They are large biological molecules with molecular weight up to few million Daltons[1]. Each protein has a well-defined function which ranges from building up DNA and RNA molecules to enzymatic catalysis, coordinated motion, signal transduction, transport, storage and immune response. Some proteins serve multiple functions.

2.2 Amino Acids

An amino acid is a molecule that contains amine (NH_2) and carboxyl (COOH) functional groups. Generally amino acids are very important in biology. They are involved in many vital activities in the bodies of living organisms as well as some industries. This involvement is summarized in the following points:

1. Amino acids play variety of roles in metabolism
2. They form parts of co-enzymes
3. Some amino acids act as precursors[2] for the biosynthesis of other molecules
4. Some other amino acids are used in food technology and industry and are generally key players in the field of nutrition

[1] Dalton: A unit of mass very nearly equal to that of a hydrogen atom. It is named after "John Dalton" (1766-1844), who developed the atomic theory of matter.

[2] Precursor: a substance from which another substance is formed (especially by a metabolic reaction)

5. α-amino acids represents the basic building block of any protein

Chemically there are several types of amino acids; α-amino acids[3] and β-amino acids[4]. Biochemistry is concerned only with α-amino acids which are called proteinogenic (those forming a protein) amino acids. α-amino acids are those amino acids whose general formula is $H_2NCHRCOOH$[5] where R is an organic substituent called the side chain. As shown from Fig 2.1, in α-amino acids both the amine and carboxyl groups are attached to the central carbon C_α. The carbon atom of the carboxyl group is called C'.

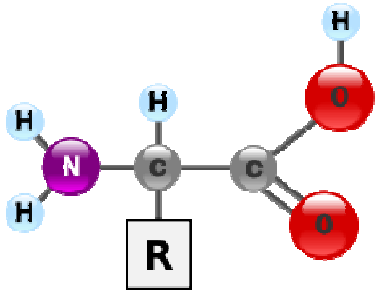


Figure 2.1: α-amino acids

What discriminates one α-amino acid from another is their side chains. A side chain can be very simple like that of *glycine* which is only one hydrogen atom or it can be a methyl group[6] like that of *alanine* or it can even be a large heterocyclic group[7] like that of *tryptophan*.

[3] α-amino acids: Any amino acid that has the amino and carboxylic functional groups attached to the same carbon atom; especially those amino acids that occur naturally as peptides and proteins

4 β-amino acids: Those which have their amino group bonded to the β carbon rather than the α carbon proteins.

[5] Proline is an exception to this formula as it contains no NH2 group

[6] Methyl Group: In chemistry, a methyl group is a hydrophobic alkyl functional group named after methane (CH4). It has the formula -CH3 and is often abbreviated -Me. Such hydrocarbon groups occur in many organic compounds.

[7] Heterocyclic Group/Molecule: An organic group/molecule containing rings with at least one non-carbon atom on the ring.

There are only 20 standard α-amino acids in nature as shown in Fig 2.2. The International Union of Biochemistry and Molecular Biology (IUBMB) and The International Union of Pure and Applied Chemistry (IUPAC) now also recommend standard abbreviations for the two additional amino acids; *L-Selenocysteine* and *L-Pyrrolysine* which are (Sec/U) and (Pyl/O) respectively.

2.3 Chemically, what is a protein?

Amino acids are linked to each other using polypeptide bonds[8] to form polypeptide chains. Around 40-50 residues appears to be the lower limit for a biologically functional peptide sequence. Although proteins' lengths start from lower limit, most of the proteins are longer than this. Actually some proteins contain thousands of amino acids e.g. membrane proteins[9]. Average protein length is around 300 residues [42].

[8] Peptide Bond: is a chemical bond formed between two molecules when the carboxyl group of one molecule reacts with the amino group of the other molecule.

[9] Membrane Protein: is a protein molecule that is attached to, or associated with the membrane of a cell or an organelle. More than half of all proteins interact with membranes.

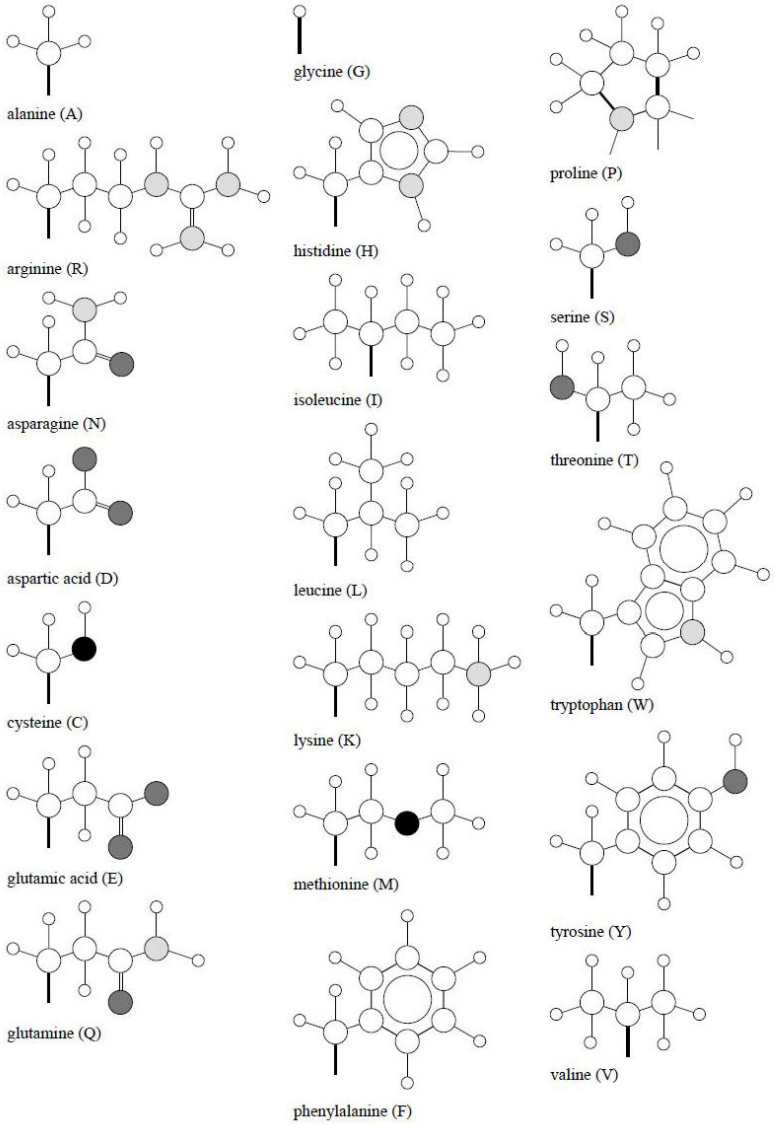


Figure 2.2: The names, abbreviations and R-groups of the twenty amino acids. Atoms are coloured white for carbon, light gray for nitrogen, dark grey for oxygen and black for sulphur. Small atoms are hydrogen. Bonds connecting R-groups to main-chain atoms are drawn in bold. Note that the proline side chain joins the main chain at both Cα and N atoms, which are shown

2.4 Diversity of proteins

For the purpose of discussing the diversity of proteins in nature, take a sequence (chain) of only 10 amino acids as an example. Theoretically the number of chains that can be formed by substitution of amino acids in

each of the 10 positions is 20^{10} (20 to the power of 10), which are approximately 10 trillion different chains. Taking into consideration longer chains will increase the number of possibilities exponentially to unbelievable limits. Although theoretical calculations yield huge figures, luckily these figures are not even close to the actual number of possibilities found in nature. The actual number of possibilities of a chain with a given length – though still big – is much smaller than the theoretically calculated figure.

Unfortunately studying proteins merely as chains of amino acids is pointless. It was found that the role of a protein is not a direct function of its constituents, however it rather depends on the three dimensional shape of its sequence in space. When a protein is in action it takes a specific conformation. This conformation is the three dimensional structure representing the final stable shape of the protein in its solvent. This final structure is called the protein *native state*. This native state is what determines the functionality.

2.5 Structural definition of a protein

In order to simplify the way of studying protein structure in its different levels of abstraction, a protein is described in four levels; primary, secondary, tertiary and quaternary, depending on the amount of structural details included in the description.

2.5.1 Primary structure

The primary structure is the simplest way of describing a protein. It specifies the structure of covalent bonds[10] in a protein molecule. So the primary structure is merely the chain of amino acids or residues[11] forming the protein. As stated previously, amino acids are linked head-to-tail

[10] Covalent Bond: is a chemical bond that involves sharing a pair of electrons between atoms in a molecule.
[11] Residue: (here) is an amino acid, which is a basic element in a protein sequence.

using peptide bonds i.e. the *C′* of amino acid residue *i* is connected to *N* atom of amino acid residue *i+1*. The convention is to list residues in head-to-tail order starting at the nitrogen atom *N* of one side terminal residue and ending at the *C′* atom of the other terminal residue. The *i-th* residue of a protein is the *i-th* residue of its sequence listed in the conventional order. It is also agreed on saying a protein of length *n* referring to a protein whose sequence contains *n* residues. See Fig 2.3.

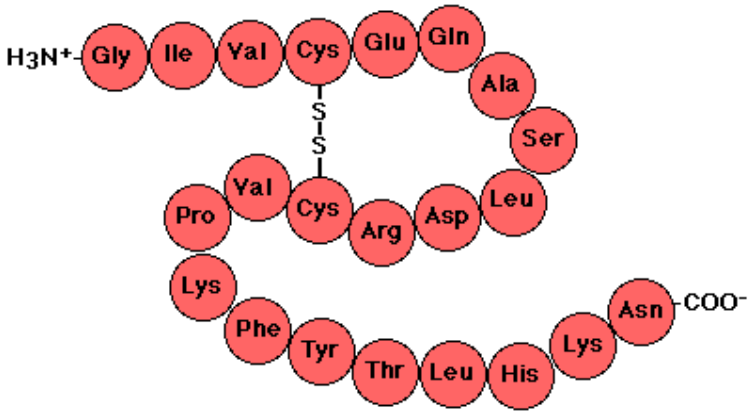


Figure 2.3: Primary structure

2.5.2 Secondary Structure

This is the higher level of conformation that describes frequent patterns in the fold of the sequence. In other words the secondary structure is an abstraction of the detailed positioning of each atom in a protein molecule using its x, y and z coordinates (which is referred to by the name "tertiary structure" later on). Instead of being concerned with the exact coordinates of each atom, the secondary structure assigns each segment of continuous residues to one of several conformational classes. Stating that a segment of *n* residues falls into one of these classes describes the contribution of each of its residues to the final local conformation of this segment. The most frequent patterns (classes) according to Kabsch and Sandler [1.3] are:

1. Helices (spiral parts in Fig 2.4)
2. Strands (arrow sheets in Fig 2.4)
3. Coils (all the other parts in Fig 2.4)

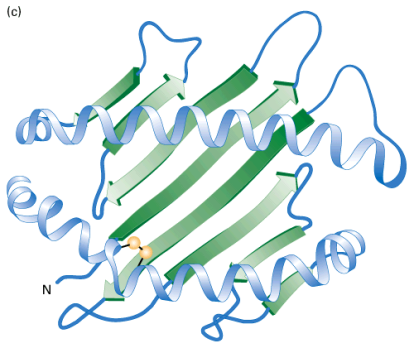
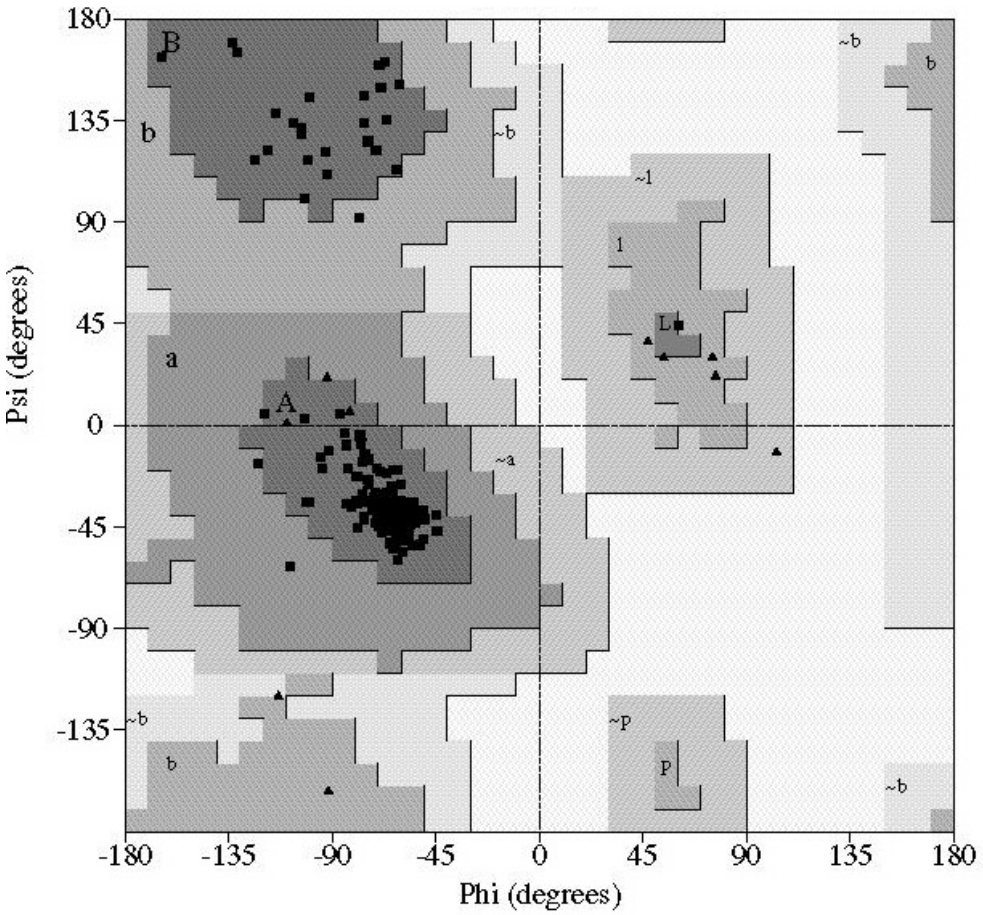


Figure 2.4: Secondary structure

2.5.3 Tertiary structure

As discussed previously, backbone residues can form an enormous number of different sequences. However this is not the only challenge. Even for a particular sequence it is hard to describe its final stable shape in 3D. This is true because even slightly different proteins (with minor differences in their sequences) can have completely different final conformations. This happens because the internal angles of the backbone are not rigid; they can rotate in almost each residue. These torsion angles are called Ramachandran angles, and by convention the angle rotating around the N-C_α bond is called ϕ while the angle rotating around the bond C_α-C' is called Ψ (See Fig 2.6). The plot shown in Fig 2.5 is called Ramachandran's plot. It highlights the most likely measurements of ϕ and Ψ.



Plot statistics

Residues in most favoured regions [A,B,L]	183	97.9%
Residues in additional allowed regions [a,b,l,p]	4	2.1%
Residues in generously allowed regions [~a,~b,~l,~p]	0	0.0%
Residues in disallowed regions	0	0.0%
Number of non-glycine and non-proline residues	187	100.0%
Number of end-residues (excl. Gly and Pro)	3	
Number of glycine residues (shown as triangles)	23	
Number of proline residues	9	
Total number of residues	222	

Based on an analysis of 118 structures of resolution of at least 2.0 Angstroms and R-factor no greater than 20%, a good quality model would be expected to have over 90% in the most favoured regions.

Figure 2.5: Ramachandran plot

Theoretically any 3D conformation can be achieved by rotating ϕ and Ψ angles. The term *"tertiary structure"* refers to the 3D description of the native fold of a protein. Backbone tertiary structure can be described

either using its n (ϕ and Ψ.) pairs or using the three dimensional spatial coordinates of each atom. A full specification of the tertiary structure requires the details of the backbone as well as the side chain which can also be described using angles or atom coordinates.

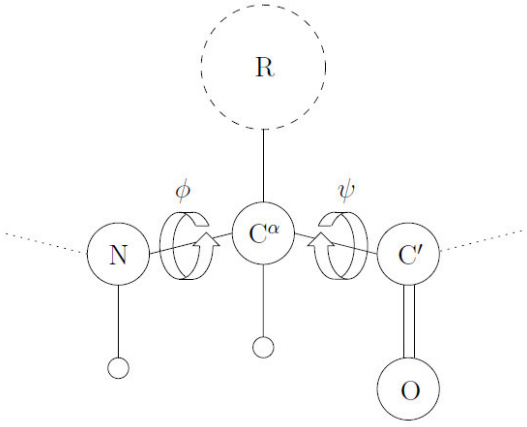


Figure 2.6: ϕ and Ψ angles

2.5.4 Quaternary structure

A quaternary structure is a collection of tertiary structures of more than one chain forming a single protein. Thus, only those proteins composed of more than one chain have quaternary structures.

2.6 The truth about protein folding

Surprisingly from all the possible conformations which are actually enormous, only about 1000 different natural protein folds are found in nature [43]. This is actually a very small number compared to the number of theoretical possibilities.

A protein can be folded (to its native state) and unfolded (to a flexible open chain) reversibly by changing the pH value, or the concentration of some denaturant[12] in solution [44]. In 1931 Wu [45] pointed out that

[12] Denaturant: a substance used for denaturation.

denaturation[13] was in fact the unfolding of a protein from *"the regular arrangement of rigid structure to irregular, diffuse arrangement of flexible open chain"*. In other words, denaturation is unfolding a protein from its tertiary structure to its primary structure.

2.6.1 How does a protein fold?

Folding actually depends on several factors. The most important factor is the hydrophobicity. Hydrophobicity is the degree to which amino acids dislike water. It was found that proteins tend to have a hydrophobic core i.e. in the folding process the hydrophobic amino acids (those hating water) tend to be in the center of the conformation and they are surrounded by a covering surface of hydrophilic amino acids (those liking water). A pioneer work by Alfinsen in the so called "Thermodynamic hypothesis" in the late 1950's and early 1960's [46] as well as other subsequent researches proved that the amino acids sequence or the so called the primary structure of the protein has all the information needed to know the complete final folding of this protein. Alfinsen stated also that the native conformation is the conformation with the global minimum free energy. This leads to what is called *"Levinthal paradox"*.

2.6.2 Levinthal paradox

"if a protein is to find its functional conformation by wandering randomly throughout conformation space, in excess of 1050 years would be required for folding" [47]. However, Levinthal and Wetlaufer pointed out that proteins fold much too fast (by at least tens of order of magnitude) to involve an exhaustive search. This is the so called Levinthal paradox. But how can a protein find a native state without a globally exhaustive search?

[13] Denaturation: Denaturation is a process in which proteins or nucleic acids lose their structure (tertiary structure) by application of some external stress or compound

The answer of this question is that since the native conformation is the global minimum of Gibbs free energy, there exists a great sort of cooperative work in protein folding. Experiments have shown that the actual work of folding is done using a very limited number of pathways guided by the thermodynamic factors. It has been observed by Sali et al. [48] that the folding time seems to be small if and only if the difference in energy between the lowest energy conformation and the next lowest energy conformation is large. Their work indicates that thermodynamic factors might have an important role in the folding process which greatly limits the number of possibilities and prevents the protein from getting lost in the huge search space.

In the following chapter some of the most popular and successful methods are discussed. Chapter 3 gives an overview of the current state of the art techniques as well as some of the previous outstanding studies in the field of protein folding. It also put this study in its place with respect to these studies.

Chapter 3
Related Work

Protein structure prediction has been one of the most challenging problems facing researchers over the few last decades. Exact prediction was found to be too far from today's state of the art even using simplified models such as the – Hydrophobic-Polar (HP) model which was found to be NP-complete[14] [1]. Several approaches have been proposed to simplify the prediction process. There are several criteria used to classify previous attempts in the field of protein folding. One of the most popular criteria used to differentiate prediction approaches from each other is the level of predictor's knowledge prior to the prediction process. Using this criterion, prediction approaches are said to be either *ab initio* or *homology* methods. *Hybrid* approaches are also available.

3.1 Homology Methods (Comparative Modeling)

So far, protein folding prediction methods based on homology have been the most successful ones. Homology modeling is based on the notion that new proteins evolve gradually from existing ones by amino acid substitution, addition, and/or deletion and that the 3D structures and functions are often strongly conserved during this process. Many proteins thus share similar functions and structures and there are usually strong sequence similarities among the structurally similar proteins.

For comparative modeling, local sequence comparison methods are usually used since the sequence similarity is most likely over segments of the two sequences. The local sequence comparison can either be pair wise or profile based. Pair wise comparisons, such as the widely used BLAST [20] in the early days, can detect sequence similarities better than 30%. A

[14] See appendix (II)

number of tools have also been developed to detect weak homology relationships. Methods like profile [21] and HMM [22] use a statistical profile of a protein family.

Since its invention in the early 1990s [27, 28], threading has become one of the most active areas in proteins structure prediction. Numerous algorithms have been developed during the past 19 years for the purpose of identifying structure templates from the PDB, which use techniques including sequence profile–profile alignments (PPAs) [29–32], structural profile alignments [33], hidden Markov models (HMMs) [34,35], machine learning [36,37], and others.

The question facing most of the researchers is: "To what extent are the currently known protein structures dependable in predicting novel and unattended structures?"

3.2 Ab Initio Methods

Unfortunately, up to this moment the answer of this question is "Not dependable enough yet". In other words, the possible protein structure permutations are still much more than those found in currently known structure, the thing that obligates researchers to use another approach that is independent of the currently known structure libraries. This family of algorithms and techniques is called "Ab Initio" methods.

When no suitable structure templates can be found, Ab Initio methods can be used to predict the protein structure from sequence information only. Conformations that minimize the energy function are taken to be the structures that the protein is likely to adopt at native conditions. In this case the most obvious representation is a number of connected points in space, each point represent an atom in the protein under investigation. Unfortunately the complexity of such a representation makes the solution simply impossible with today's computational capacity. Even if the

representation is further simplified and the points represent whole residues instead of atoms, the problem remains intractable. So – for practical reasons – most Ab Initio prediction methods use reduced representations of the protein to limit the conformational space to manageable size and use empirical energy functions that capture the most important interactions that drive the folding of the protein sequence toward the native structures.

Of course ab initio techniques are not as successful as homology techniques that use existing structures as a guidance through the prediction process. However there are other strong motivations to pursue research in this field:

1. Ab initio techniques are the last resort for discovering unattended structures.
2. They give insights on the internals of the folding process.
3. They can be applied – outside of the prediction problem – to the large scale conformational change in protein functioning.

One of the best Ab Initio methods is ROSETTA which performed better than other Ab Initio methods in CASP. Actually most of the pure ab initio approaches are now obsolete because it was fond that even with ab initio approaches it is very useful to use some sort of prior knowledge extracted from existing protein structures.

3.3 Hybrid Methods

Most of the recent approaches tend to use hybrid double-staged prediction algorithms [16]. In this type, the output of the first stage is taken as an input into the second stage. In many cases the first stage is used to approximately predict the secondary structure of a protein [9] while the second stage continues to approximately predict the tertiary structure [16]. Due to the difficulty of the folding problem, researchers use

approximated and simpler protein representations like using lattice structures such as HP models [9] and face-centered-cubic lattice [8,16] and/or using heuristic techniques in order to simplify the calculations [2,7,17].

3.4 Overview of the different approaches

Using subsequence structural information as a step towards the ultimate goal of complete prediction is widely used in literature [4,6,10,14,15]. Another approach is to deal with the protein as a whole [7,11,12,13,16,17] trying to find the optimum conformation with minimum free energy [11]. Statistical analysis of protein subsequences has appeared in literature too. Rong She et al. used two types of subsequence classifiers to identify outer membrane proteins of Gram-negative bacteria [14]. Eran Segale and Daphne Koller introduced a general probabilistic framework for clustering biological data into a hierarchy [5]. Eli Hershkovitz et al. used torsion angles to search for clusters in RNA conformational space [4]. Estimating the probability density function was used by Diego Rother et al. with the notion of ensembles [3]. Marcio Dorn and Osmar Norberto used the ϕ and Ψ angles of the central residue of a subsequence along with a secondary structure prediction method to cluster subsequences (fragments). The following sections categorize and explain these attempts – as well as some others – in greater detail.

3.5 Using lattice models

3.5.1 Lattice models, why and how?

As previously mentioned the number of possibilities to which a protein can fold is very large making the exhaustive search throughout the search space almost impossible (at least using today's most powerful supercomputer). This leads by the researchers to use discrete lattice

structures where each node in the lattice represents one amino acid residue. This is an approximated model of the protein conformation. Of course discrete lattice models greatly simplifies the search space, however this comes with a cost. Using discrete lattice models leads to the loss of many of the specific details of the actual protein conformation. The advantages of using discrete models can be summarized in the following point:

1. Eliminating a vast area of the search space
2. The ability to model large number of conformational changes. This is not possible with detailed atomic models
3. Reducing the number of parameters used in order to solve the problem. Actually this is a direct result of eliminating the conformation fine details.
4. Representing a prediction of the overall conformation of large polypeptide chains such as proteins, the thing that is time consuming and almost impossible with detailed atomic models.

On the other hands the disadvantages are:

1. These models limit the angles between any three consecutive amino acids of the backbone to specific measurements. For example square and cubic HP models limit all the angles to 90". Other models allow more measurements such as 45", 135", 120" and others. This is a great pitfall since the actual measurements are not so confined.
2. Some details are eliminated such as bond energies and charges
3. Reaching a stable structure in a lattice model can be different from the actual stable structure of the protein.

Even discrete lattice models have several variations. One of the most famous models is the HP model introduced in a pioneering work by Dill [49].

3.5.2 HP model

In this type of models a primary structure with *n* amino acids is viewed as a sequence S = <s_1, s_2 … s_n>. The symbol s_i represents an amino acid residue. Each residue can be either hydrophobic or hydrophilic. The legal conformations are self-avoiding paths on a lattice, usually taken to be Cartesian (sometimes triangular lattice), in which vertices are labeled by the amino acids.

This model depends on the hydrophobicity of each amino acid. Two adjacent amino acids in the conformation that are not successive in the sequence add *-1* to the overall energy value. In this way the conformation with most contiguous hydrophobic amino acids i.e. the one with a hydrophobic core will have the smallest energy value.

HP lattice can be viewed in 2D or 3D forms. See Fig 3.1, Fig 3.2 and Fig 3.3 respectively. In 2D lattice all the torsion angles are 0". In the 3D lattice a torsion angle is either 0" or 180". In fact this lattice is very simple and ignores many features and details of the conformation, but it captures the major details of the fold.

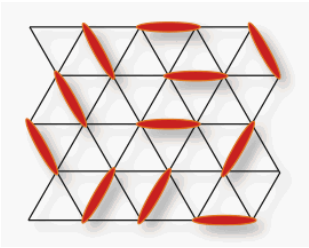


Figure 3.1: Triangular HP model

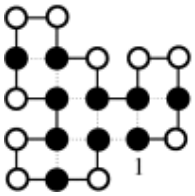


Figure 3.1: 2D HP square lattice

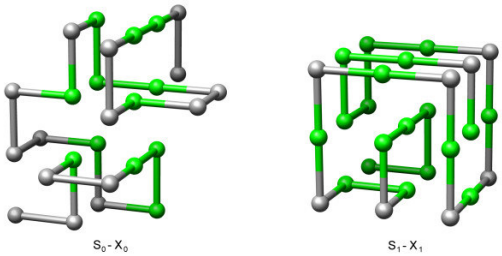


Figure 3.2: 3D HP square lattice

3.5.3 FCC model

The HP model was extensively studied in literature however there is a debate in its usefulness because it is considered too far from the required accuracy. Other models have been introduced. A face-centered cubic lattice called cube-octahedron that has 14 faces and 12 vertices was introduced by Raghunathan and Jernigan in 1997 [8].

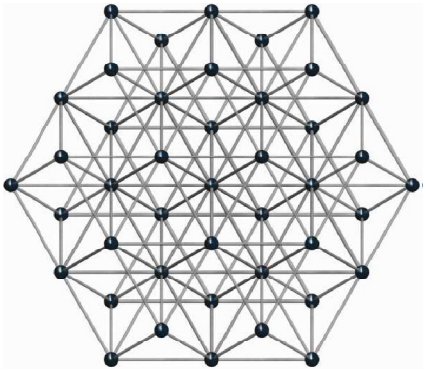


Figure 3.3: Cube-Octahedron lattice

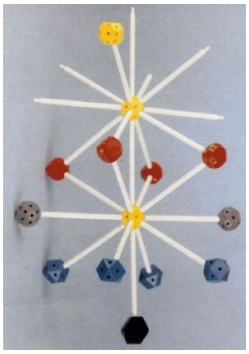


Figure 3.4: Angles of cube-octahedron lattice

As a face-centered cubic lattice all the vertices has the same distance from the center. Fig 3.4 represents the shape of the lattice. Assume that each atom is a C_α. As seen in Fig 3.5 the possible angles between each three connected C_α are 60", 90", 120" and 180". In Fig 3.5 the red balls makes a 60" with the vector made by the two yellow nodes connecting the two units represented in the figure. The two grey balls make an angle of 180" with the same vector. Although there are several angle choices in this lattice the only allowed angle measurements in a protein are 90" and 120" due to steric constraints[15]. Consequently, three of such vectors can define valid torsion angles, typically the measurements of these torsion angles are 54.7", 109.5", 125.3" and 180".

Cube-octahedron is much more flexible than the HP model (even its 3D form). It also gives a more realistic representation to secondary structure motifs (α-helices and β-strands).

3.6 Using heuristic techniques

Various optimization methods have been applied to formulations of the folding problem – especially with ab initio methods – that are based on

[15] Steric constraints: Steric effects arise from the fact that each atom within a molecule occupies a certain amount of space. If atoms are brought too close together, there is an associated cost in energy due to overlapping electron clouds.

reduced models of protein structure, including Monte Carlo methods, Genetic algorithms, Tabu Search, simulated annealing, hybrid techniques among others.

In 2005 Thang N. Bui and Gnanasekaran Sundarraj presented an efficient genetic algorithm for the protein folding problem under the HP model in the two-dimensional square lattice [17]. A special feature of this algorithm is its usage of secondary structures, which the algorithm evolves, as building blocks for the conformation. Experimental results on benchmark sequences show that the algorithm performs very well against previously known evolutionary algorithms and Monte Carlo algorithms.

In the same year – 2005 – Alena Shmygelska1 and Holger H. Hoos used introduced an ant colony optimization (ACO) algorithm to address the non-deterministic polynomial-time hard (*NP*-hard) combinatorial problem of predicting a protein's conformation from its amino acid sequence under a widely studied, conceptually simple model – the 2-dimensional (2D) and 3-dimensional (3D) hydrophobic-polar (HP) model [7]. This is an improvement of their previous ACO algorithm for the 2D HP model and its extension to the 3D HP model. The empirical results they got indicate that their rather simple ACO algorithm scales worse with sequence length but usually finds a more diverse ensemble of native states.

In 2006 Clayton Matthew Johnson and Anitha Katikireddy proposed a simple genetic algorithm for finding the optimal conformation of a protein using the three-dimensional square HP model [2]. A backtracking procedure is used to resolve the positional collisions and illegal conformations that occur during the course of genetic search. Backtracking is shown to be a simple and efficient means of collision repair that requires little overhead. Empirical results show that a genetic

algorithm using backtracking can obtain the lowest energy structure of an amino acid sequence in fewer energy evaluations than earlier approaches.

In 2007, Xiaolong Zhang, Xiaoli Lin, Chengpeng Wan and Tingting Li introduced a genetic algorithm for 3D off-lattice protein folding [40]. In this approach the PSP problem is converted from a nonlinear constraint-satisfied problem[16] to an unconstrained optimization problem. They showed that their approach have promising results in three dimensional prediction.

In 2008, Madhu Smita, Harjinder Singh and Abhijit Mitra used a variant of standard genetic algorithms called real valued genetic algorithm in order to solve the PSP problem [39]. In the same year, Xiaolong Zhang and Wen Cheng proposed an algorithm that uses an enhanced version of Tabu Search (TS) for 3D off-lattice protein folding. They claim that their approach successfully reaches conformations with a single hydrophobic core which makes these conformations more realistic than those developed by previous methods [41].

3.7 Use of subsequence structural information

Many researchers used information of the subsequences of a protein or a peptide chain in order to predict the whole structure of the protein.

In [15] Saravanan Dayalan et al. proposed a dihedral angle database of short sub-sequences up to length 5. They claimed that the proposed database would handle protein structure prediction program queries efficiently that is based on short subsequences and exact matches. Previously proposed dihedral databases have limitations such as not being able to retrieve dihedral values for one or more amino acids occurring in sub-sequences or designed for a specific set of proteins based on its sequence identity. The database proposed in this paper overcomes these

[16] Constraint-Satisfied Problem (CSP): See appendix (III)

limitations by considering all proteins of PDB during dihedral angle extractions and by extracting dihedral values of one or more amino acids that occur in a specific sub-sequence.

In 2005, Hardik A. Sheth and Sun Kim aligned protein subsequences and clustered them in order to discover similar motifs[17]. They used clustering in order to generate clusters of homogenous sequences when the input sequences are non-homogenous [6].

Another attempt made in [10] was to use torsion angles measurement of the subsequences in order to predict secondary structures (discussed later in detail).

3.8 Dealing with the protein as a whole

Many researchers look at the protein as a whole instead of considering its subsequences. Most of the approaches falling in this category are those that use heuristic techniques.

In 1997 Richa Agarwala et al. presented a set of folding rules for a triangular lattice and analyze the approximation ratio which they achieve [13]. They also tried to compare several lattice structures and choose the best. They claimed that the best choice is the triangular lattice.

In [7] – as discussed previously – an ACO algorithm is applied to the protein conformation as a whole.

In 2003, Neal Lesh et al. presented new lowest energy configurations for several large benchmark problems for the two-dimensional hydrophobic/hydrophilic model. They found these solutions using tabu search using a novel set of transformations that they called pull moves. Their experiments show that their algorithm can find these best solutions

[17] Motifs: are short, conserved subsequences that are part of a family of subsequences. The use of protein sequence patterns (or motifs) to determine the function of proteins is an essential tool for sequence analysis.

in 3 to 14 hours, on average. Pull moves appear quite effective and may also be useful for other local search algorithms for the problem [12].

As discussed in [17], a genetic algorithm was introduced to solve the problem in 2D HP lattice.

In 2007, one of the most remarkable studies was carried out by Sergio Raul Duarte Torres et al. In this study a model based on genetic algorithms for protein folding prediction is proposed. The outline is depicted in Fig 3.6. The most important features of the proposed approach are:

1. Heuristic secondary structure information is used in the initialization of the genetic algorithm in order to generate realistic – with realistic structures – chromosomes for the initial random population.
2. An enhanced 3D spatial representation called cube-octahedron is used (review section: using lattice models - FCC model).
3. Data preprocessing of geometric features is made to characterize the cubeoctahedron using twelve basic vectors to define the nodes.
4. Biological information (torsion angles, bond angles and secondary structure conformations) was pre-processed through an analysis of all possible combinations of the basic vectors which satisfy the biological constrains defined by the spatial representation.
5. Hashing techniques were used to improve the computational efficiency. The pre-processed information was stored in hash tables, which are intensively used by the genetic algorithm.

The implementation developed in this research drastically decreased the algorithmic complexity of the protein folding construction and search. Specifically, strategies such as data preprocessing, hashing techniques and spatial vector representation made possible a highly efficient model in terms of time and computational resources.

The use of hash tables provides an excellent computational technique to model amino acid spatial occupancy, because the number of collisions are reduced to zero and the insertion, erasing and search are very efficient.

Secondary structure information is fundamental for the accuracy of the predicted models, given the importance of those conformations in the protein folding process present in nature [16].

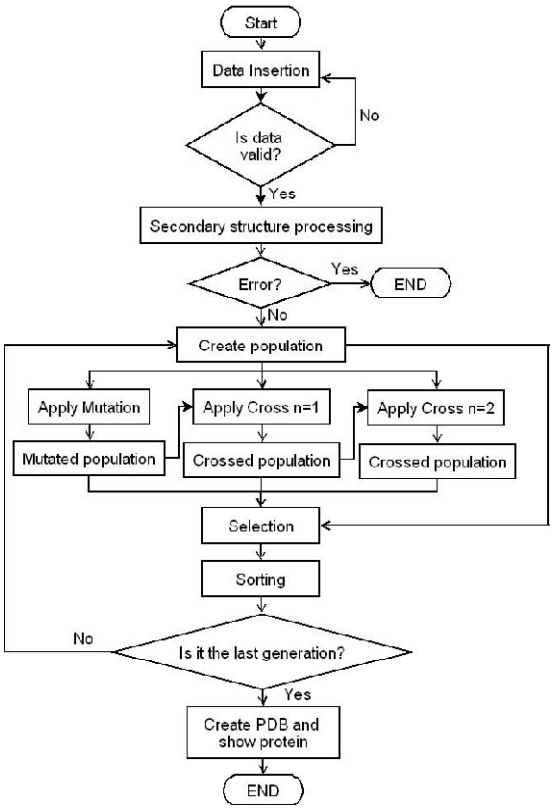


Figure 3.5: Outline of the algorithm proposed in [16]

3.9 Statistical analysis of protein subsequences

Using statistical analysis in bioinformatics is widely used in literature. In 2002, Eran Segal and Daphne Keller introduced Probabilistic Abstraction Hierarchies (PAH) a general probabilistic framework for clustering data into a hierarchy [5]. Biological data, such as gene expression profiles or protein sequences, is often organized in a hierarchy of classes, where the instances assigned to “nearby” classes in the tree are similar. Most

approaches for constructing a hierarchy use simple local operations, that are very sensitive to noise or variation in the data. In their work Eran and Daphne showed how PAH can be applied to gene expression data, protein sequence data, and HIV protease[18] sequence data. This feature helps in avoiding local maxima and in reducing sensitivity to noise.

In 2006, Eli Hershkovitz et al. tried to solve the problem of predicting RNA conformations [4]. Predicting RNA conformation is more complex than predicting protein structure due to the large number of degrees of freedom (torsion angles) per residue. In their work, they used and extended classical tools from statistics and signal processing to search for clusters in RNA conformational space.

3.9.1 Estimating probability density function

In 2008, Diego Rother et al. made a remarkable attempt that worth mentioning. In one of the very few times they tried to study the fluctuation and variation under physiological conditions [3]. In their work they introduced a framework for estimating probability density functions in very high dimensions and then apply it to represent ensembles[19] of folded proteins. Although this is not strongly related to the approach pursued in our study, using probability density functions is the common aspect between the two studies.

3.10 Prediction using angle measurements

Most of the researcher who worked on the protein folding problem – especially those who didn't use lattice models – studied the two angles ϕ and Ψ. One of the most recent approaches is that introduced by Márcio

[18] Protease: any enzyme that catalyzes the splitting of proteins into smaller peptide fractions and amino acids by a process known as proteolysis.

[19] Ensemble: An ensemble is a set of conformations of the same protein. Each conformation corresponds to a particular arrangement of the protein's constitutive atoms in three-dimensional (3D) space. This arrangement can be described (or partially described) by different sets of features depending on the application at hand. In [3] the backbone of the protein is described by the usual torsion angles ϕ and Ψ.

Dorn and Osmar Norberto de Souza in 2008. In their study they defined what is abbreviated as "CRef" (a central-residue-fragment-based method). With CReF they expected to obtain approximate 3-D structures which can then be used as starting conformations in refinement procedures employing state-of-the-art molecular mechanics methods such as molecular dynamics simulations. CReF does not make use of entire fragments, but only the ϕ and Ψ torsion angle information of the central residue in the template fragments obtained from PDB. After applying clustering techniques to these data they built approximated conformations for the target sequence. Their method is very fast. Their initial results show that the predicted conformations adopt a fold similar to the experimental structures.

They applied their approach to three case studies – three peptide chains – with lengths ranging from 34 to 70 amino acid residues. In the three case studies their approach reached correct secondary structures and overall fold predictions. For longer chains the results were less accurate. CRef failed to predict correct tertiary structures due to its limited ability to predict coiled parts such as turns and loops.

3.11 How this study differs from prior art?

Simply this study is an attempt to introduce an alternative approach to be used in the first stage of hybrid techniques of solving the PSP problem. The approach proposed herein this study tries to find the best probability distributions which fit the angles measurements of clustered subsequences. Clustering uses hydrophobicity as a similarity function. These fits are then analyzed statistically to determine the effect of hydrophobicity and subsequence length on backbone angles. Chapter four explains the approach in detail.

Chapter 4

A Central-3-Residues-Based Clustering Approach for Studying the Effect of Hydrophobicity on Protein Backbone Angles

4.1 Approach Outline

As discussed in chapter 3, this study tries to find a relation between the central angle of each *n*-amino acids and the hydrophobicity of these surrounding amino acids. The problem is tackled in a four-phased approach. The four phases are:

1. Angle extraction
2. Chopping
3. Clustering
4. Distribution fitting

Fig 4.1 shows the four phases. The first two phases are data preparation phases. The input of the whole system is SCOP[20] entries[21].

Phase 1 calculates the central angles of all the subsequences contained in each entry using the x, y and z coordinate measurements of each atom which are available in the SCOP entries this step is important because each SCOP entry contains a huge amount of information about the protein. All what is of concern to this study are the 3D coordinates of the backbone atoms which are used to calculate the backbone angles measurements (as they are not explicitly stored in the SCOP entries).

Phase 2 is called chopping i.e. dividing each protein into subsequences. These subsequences are overlapping as will be shown later.

[20] SCOP: is a structural classification of *proteins database* for the investigation of sequences and structures.
[21] SCOP entry: A file representing a single protein in SCOP database

The third phase clusters protein subsequences using their hydrophobicity as the similarity function. The approach used in this study is applied to the test data once before clustering and once after clustering. Clustering creates groups of subsequences of similar hydrophobicity patterns. On the other hand, ignoring the clustering step – i.e. dealing with the whole data set as a single cluster – groups all the subsequences in a single set even if there hydrophobicity patterns were so different. If the relationship is more evident in the big single cluster than it is in the small clusters, we can deduce that clustering is useless and that the hydrophobicity pattern of the subsequence has nothing to do with the central angle measurement. However if the small clusters show more evident relationships then we can say that hydrophobicity patterns has direct impact on the central angle measurements. Fig 4.1 shows that the third phase (clustering) is optional, in this way it is possible to assess the effect of clustering on the final relationship between angle measurement and hydrophobicity.

The fourth phase applies the KS test to find out the best continuous probability distribution that fits the central angle measurements of each cluster (or of the whole test data set if clustering is ignored). The KS-test generates two types of statistics:

1. The KS-statistic
2. Number of rejected values

These statistics will be used in chapter 5 to discuss the results.

4.2 SCOP databank (test data set)

In this study we used a sample of 1089 proteins randomly selected from the SCOP protein databank. Each protein consists of about 200 subsequences in average, thus the total number of subsequences is more

than 200,000 subsequences. These subsequences represent the input of the system. They are then clustered as discussed earlier.

SCOP uses the same format as that of the PDB[22]. Each file in the SCOP databank represents one protein. Each record (line) in each file starts with a word indicating its type. In this study we are concerned with *atom* records. See Fig 4.2. These are records starting with the keyword ATOM. Each atom record represents one atom in the protein. Each atom record contains 15 pieces of information. Of these 15 the most important are:

1. A sequential number for each atom
2. Type label indicating the type of the atom (e.g. C_α atom)
3. Three letter abbreviation name of the residue containing the atom
4. Residue sequence number
5. Orthogonal coordinates for X in Angstroms
6. Orthogonal coordinates for Y in Angstroms
7. Orthogonal coordinates for Z in Angstroms
8. Occupancy
9. Charge on the atom

[22] PDB: The Protein Data Bank (PDB) is a repository for the 3-D structural data of large biological molecules, such as proteins and nucleic acids. (See also crystallographic database). The data typically obtained by X-ray crystallography or NMR spectroscopy and submitted by biologists and biochemists from around the world (freely accessible through the Internet).

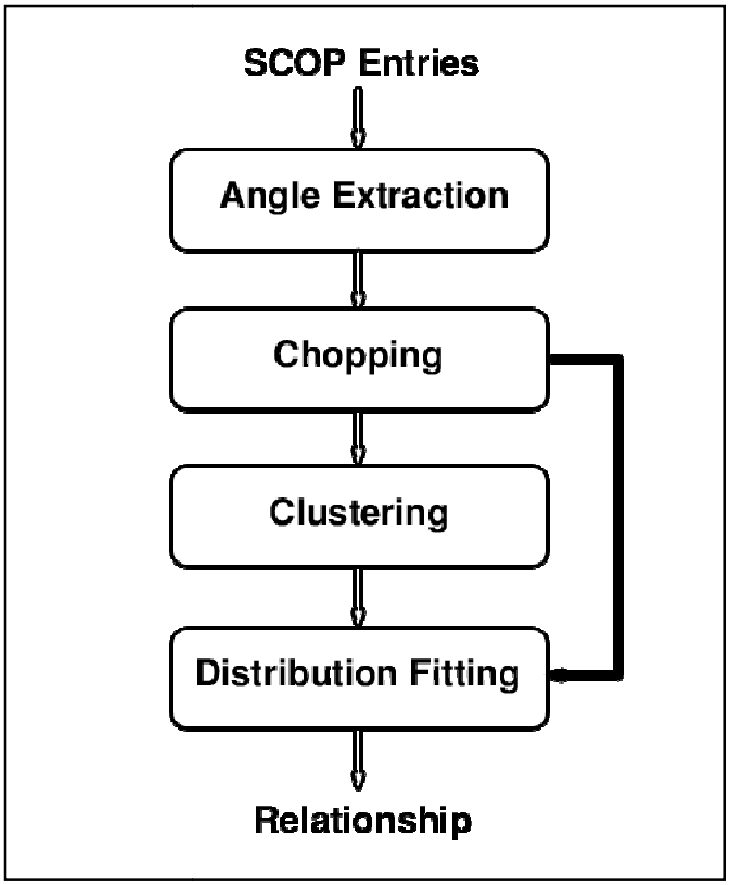


Figure 4.1: System phases

In this study we are concerned with 3, 5, 6 and 7. For a full specification of the PDB format see [50].

```
ATOM      6  CG  PRO A   2      27.370  22.419  39.983  1.00 38.52
```

Figure 4.2: SCOP/PDB ATOM record

4.3 Phase 1: Angle Extraction

In this phase all the input entries are scanned starting at the first amino-acid residue of the backbone and ending at the last amino-acid residue. The angles between each three consecutive residues are calculated. Assume that the backbone contains only 5 amino acid residues (just for

the sake of explanation). 3 theta values will be calculated; θ_1 lies between C_α^1- C_α^2- C_α^3, θ_2 lies between C_α^2- C_α^3- C_α^4 and finally θ_3 lies between C_α^3- C_α^4- C_α^5. Thus, a protein composed of *n* amino-acid residues has *n-2* theta angles.

4.3.1 How θ is calculated

Assume that we are calculating the angle lying between the three atoms C_α^{i-1}, C_α^{i} and C_α^{i+1} respectively, such that:

$$C_\alpha^{i-1} = (x_{i-1}, y_{i-1}, z_{i-1})$$

$$C_\alpha^{i} = (x_i, y_i, z_i)$$

$$C_\alpha^{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$$

The angle theta (θ) is the angle between the two vectors **a** and **b**, such that:

$$\boldsymbol{a} = (C_\alpha^i, C_\alpha^{i-1})$$

$$\boldsymbol{b} = (C_\alpha^i, C_\alpha^{i+1})$$

Theta is then calculated using Cosine law:

$$\cos\theta = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

Where the numerator is called the dot product[23] of the two vectors and the symbols |**a**| and |**b**| that appear in the denominator are called the norm[24]s of vectors **a** and **b** respectively.

[23] The dot product a.b of two vectors $a = (x_1, y_1, z_1)$ and $b = (x_2, y_2, z_2)$ is calculated by the formula: $A.B = x_1x_2 + y_1y_2 + z_1z_2$

[24] The norm of vector a = (x,y,z) is calculated by the formula: $\sqrt{x^2 + y^2 + z^2}$

4.3.2 Representation

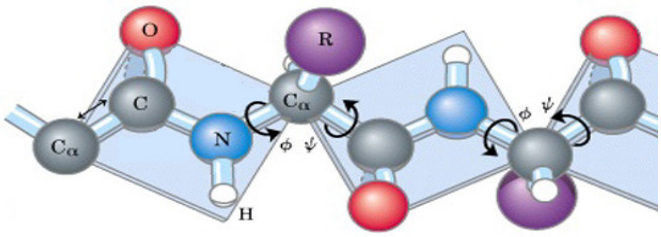


Figure 4.3: ϕ and Ψ torsion angles

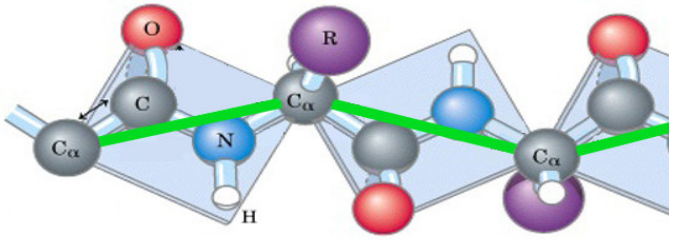


Figure 4.4: Θ-angles

A subsequence of residues is represented by a vector *(v)*. As discussed in chapter 2, each residue contains three main consecutive atoms; a central Carbon atom *(C_α)* connected to another Carbon atom *(C)*, a Nitrogen atom *(N)* and a side chain. Each amino acid contains two torsion angles; ϕ and Ψ as shown in Fig 4.3. This study is not concerned with these torsion angles however the main concern is the angle *Θ* which is the angle between the three consecutive C_α atoms of the three central residues of the subsequence (each C_α atom represents the center of one amino-acid residue). *Θ* is the angle between each two lines connecting C_α atoms in Fig 4.4. Thus a subsequence *S* is represented by a vector *v* and an angle *Θ*:

$$S = (v, \theta : v = \langle aa_{p,} aa_{p+1} \dots aa_{p+n-1} \rangle) \quad (1)$$

Where p is the starting position of the subsequence and aa_p represents the amino-acid residue at position p. Notice that the angle we are talking about here is neither ϕ nor Ψ angles of the central amino-acid. Alternatively a single (distinct) angle is taken to represent the relative positions of every three consecutive amino-acids. A centroid in this study is represented by a simple vector of n hydrophobicity values:

$$C = \langle h_{0,} h_1 \dots h_{n-1} \rangle, \quad (2)$$

$such\ that\ h_i\ is\ the\ hydrophobicity\ of\ residue\ i\ in\ the\ protein$

4.4 Phase 2: Chopping

All proteins are divided into subsequences of amino acid residues of length n. A protein of length L is divided into $L - n + 1$ subsequences starting at $(aa_0 \dots aa_{n-1})$ and ending at $(aa_{L-n} \dots aa_{L-1})$. Therefore the total number of subsequences considered in this study is $\sum_{i=0}^{N}(L_i - n + 1)$ where N is the total number of proteins. The angle Θ is the angle between the three amino acid residues in the center of the subsequence $(aa_{\frac{p+n}{2}}, aa_{\frac{p+n+2}{2}}, aa_{\frac{p+n+4}{2}})$, where p is the start position of the subsequence in the whole protein sequence. The subsequences are overlapping i.e. every two consecutive subsequences of length n shares $n - 1$ residues. The value of Θ is calculated using the coordinates of C_α atoms of these residues taken from the SCOP database as previously mentioned. Obviously the number of residues in a subsequence must be odd so that the number of residues on both sides of the angle is the same. Typical values of n used in this study are 3, 5 and 7. Higher values of n are possible but they are computationally intensive.

4.5 Phase 3: Clustering

In this phase the subsequences extracted from the chopping phase are clustered. A k-means clustering algorithm is used to cluster these

subsequences. The similarity function used in clustering is the key factor of this approach. Each amino acid residue has a hydrophobicity value as discussed in chapter two. The 20 amino acid residues available have hydrophobicity values ranging from Arginine – the hydrophilic extreme – to Isoleucine – the hydrophobic extreme – with typical values -4.5 and +4.5 respectively. Subsequences A and B are considered similar if the differences between the hydrophobicity values of each two corresponding amino acid residues in the two subsequences are small.

The Following points explain how the k-means clustering algorithm is applied:

1. Before the clustering algorithm starts, a copy of the unclustered subsequences is saved. Phase 4 – distribution fitting – will be applied to the unclustered subsequences as well as clustered subsequences in order to assess the importance of clustering in the approach under study.
2. The sets of subsequences generated from step 1 are then fed to the k-means clustering algorithm. As mentioned in phase 2, the three values of *n* considered in this study are 3, 5 and 7 i.e. three groups of subsequences are available. The clustering algorithm is applied three times to the three groups of subsequences. This algorithm uses residues hydrophobicity as a similarity measurement (as will be discussed later).
3. Initial centroids are pre-known and are based on the value of *n*. When creating the initial centroids each position in the subsequence is assumed to be either hydrophobic[25] (*H*) or hydrophilic[26] (*P*)[27]. Taking the two extremes of hydrophobicity into consideration,

[25] Having a strong aversion for water
[26] Having a strong affinity for water
[27] *P* here stands for *"polar"* which has the same meaning as *"hydrophilic"*.

namely *Isoleucine* (*+4.5*) and *Arginine* (*-4.5*), leaves us with only two choices for each residue position. Calculating all the permutations of a subsequence of length n results in a total of 2^n centroids. For example table 4.1 lists the initial centroids if $n = 3$.

Table 4.1: listing of the initial centroids used in clustering given that n = 3. Each cell represents the hydrophobicity of one hypothetical amino acid residue in the each centroid. H represents +4.5 while P represents -4.5.

H	H	H
H	H	P
H	P	H
H	P	P
P	H	H
P	H	P
P	P	H
P	P	P

4. Similarity function: Let the hydrophobicity of a residue aa be $aa.h$. The similarity function measures how a subsequence *S* is similar to some centroid *C* in terms of hydrophobicity. The function simply calculates the average of differences in hydrophobicity between the residues of *S* and the corresponding hydrophobicity values in *C*.

$$\sum_{i=0}^{n} \frac{(aa_i.h - h_i))}{n} \quad (3)$$

4.6 Phase 4: Distribution Fitting

2^n clusters of similar subsequences – in terms of hydrophobicity – resulting from the k-means clustering are obtained. Two Kolmogrov-

Smirnov (KS)[28] tests are performed. The first test is performed on the unclustered subsequences while the second test is performed on the 2^n clusters generated from phase 3. KS test is performed against the following 66 standard continuous probability distributions: Beta, Burr, Burr (4P)[29], Cauchy, Chi-Squared, Chi-Squared (2P), Dagum, Dagum (4P), Erlang, Erlang (3P), Error, Error Function, Exponential, Exponential (2P), Fatigue Life, Fatigue Life (3P), Frechet, Frechet (3P), Gamma, Gamma (3P), Gen. Extreme Value, Gen. Gamma, Gen. Gamma (4P), Gen. Logistic, Gen. Pareto, Gumbel Max, Gumbel Min, Hypersecant, Inv. Gaussian, Inv. Gaussian (3P), Johnson SB, Johnson SU, Kumaraswamy, Laplace, Levy, Levy (2P), Log-Gamma, Log-Logistic, Log-Logistic (3P), Log-Pearson 3, Logistic, Lognormal, Lognormal (3P), Nakagami, Normal, Pareto, Pareto 2, Pearson 5, Pearson 5 (3P), Pearson 6, Pearson 6 (4P), Pert, Phased Bi-Exponential, Phased Bi-Weibull, Power Function, Rayleigh, Rayleigh (2P), Reciprocal, Rice, Student's t, Triangular, Uniform, Wakeby, Weibull and Weibull (3P). Parameters are estimated using Maximum Likelihood Estimation (MLE)[30].

Results are discussed in the next chapter.

[28] Kolmogrov-Smirnov: A non-parametric statistical test used to determine if two separate samples could have been drawn from the same population, or populations with the same distributions. Or measures to how extent a standard probability density function fits a sample or a population.

[29] *2P*, *3P* and *4P* refer to two, three and four parameters distributions respectively. A typical distribution is said to be *(n-1) P* if its location parameter is set to 1 and *(n) P* otherwise.

[30] Maximum Likelihood Estimation (MLE): a method of parameter estimation in which a parameter is estimated to be that value for which the data are most likely.

Chapter 5
Results

This chapter discusses the results reached during conducting this study. The hypothesis we are trying to prove is stated. The procedure followed to generate the results is explained in detail, the tools used are listed and the results are discussed.

5.1 Hypothesis

Through conducting this study we try to argue about two assumptions:

1. The first part of the hypothesis suggests that the angles measurements of a protein sequences follow some sort of pattern based on the hydrophobicity of the surrounding local amino acid residues.
2. The second part suggests that the reliability of these patterns increases as the number of neighboring amino acid residues taken into consideration increases.

5.2 Procedure

As discussed in chapter 4 the procedure consists of 4 consecutive steps. Here we are going to re-order and thoroughly explain them to indicate the actual sequence of the detailed steps of the algorithm in the following 15 points:

1. Angle extraction[31]
2. Chopping[32]
3. Apply a KS-test on **unclustered** subsequences of length $n = 3$ [33]
4. Apply a KS-test on **unclustered** subsequences of length $n = 5$ [34]

[31] For a complete discussion of this step, refer back to section 4.3
[32] For a complete discussion of this step, refer back to section 4.4
[33] For a complete discussion of this step, refer back to section 4.6

5. Apply a KS-test on **unclustered** subsequences of length $n = 7$ [35]
6. K-means Clustering[36]
7. Apply a KS-test on **clustered** subsequences of length $n = 3$ [37]
8. Apply a KS-test on **clustered** subsequences of length $n = 5$ [38]
9. Apply a KS-test on **clustered** subsequences of length $n = 7$ [39]
10. Compare the KS-statistic values of the tests carried out in steps 7, 8 and 9 (in order to find the effect of increasing the value of *n* with **clustered** subsequences).
11. Compare the number of rejected values (out of five values) of the tests carried out in steps 7, 8 and 9 (In order to determine the reliability of the fits when using **clustered** subsequences).
12. Compare the KS-statistic values of the tests carried out in steps 3, 4 and 5 (in order to find the effect of increasing the value of *n* with **unclustered** subsequences).
13. Compare the number of rejected values (out of five values) of the tests carried out in steps 3, 4 and 5 (In order to determine the reliability of the fits when using **clustered** subsequences).
14. Compare the results of steps 10 and 12 (In order to find out the effect of clustering).
15. Compare the results of steps 11 and 13 (In order to find out the effect of clustering).

[34] For a complete discussion of this step, refer back to section 4.6
[35] For a complete discussion of this step, refer back to section 4.6
[36] For a complete discussion of this step, refer back to section 4.5
[37] For a complete discussion of this step, refer back to section 4.6
[38] For a complete discussion of this step, refer back to section 4.6
[39] For a complete discussion of this step, refer back to section 4.6

5.3 Tools

Two pieces of software[40] have been developed to perform the first three steps:

1. The first is the *preparation engine* which is fed with the SCOP entries on which it performs angle extraction and chopping.
2. The second is the clustering engine which is responsible for applying the k-means clustering algorithm on the chopped subsequences.

Distribution fitting (KS-test) is performed using a readymade software package called *EasyFit*.

5.4 Results

First of all hydrophobicity values of the final centroids are listed for each value of *n* (remember that *n* is the subsequence length). The results of the k-means clustering are then discussed as well as the results of the KS tests for both clustered and unclustered subsequences.

Table 5.1 summarizes the best fitting distributions for all the three values of *n*. The goodness of these fits will be discussed later in this section. The left column contains the name of continuous probability distributions while the corresponding cells in the right column contain the IDs of the centroids that fit into these distributions i.e. the third row in table one indicates that the two centroids $C_1$and C_4 resulting from clustering subsequences of length 3 fits into Burr continuous probability distribution. The typical hydrophobicity values of these centroids are listed in appendix (I).

[40] Developed in Java

Table 5.1: Best fitting distribution of each centroid of the three values of *n*

continuous distribution	centroids that the continuous distribution best fits
n = 3	
Burr	1, 4
Burr(4p)	7
Gen. Extreme Value	6
Gen. Pareto	2, 3, 5
Johnson SB	0
n = 5	
Dagum(4p)	0, 5, 7, 19
Gumbel Min.	1, 2, 3, 17, 20
Gen. Extreme Value	4, 32
Burr(4p)	6, 8, 10, 11, 14, 18, 21, 22, 23, 24, 27, 30, 31
Weibull(3p)	9, 12, 13, 15, 16, 25, 26, 28, 29
n = 7	
Weibull(3p)	3, 21, 79
Burr(4p)	20, 32, 40, 60, 67, 71, 74, 75, 83, 85, 105
Dagum	4, 80
Dagum(4p)	41, 90
Gen. Gamma(4p)	69, 84, 106
Gen. Logistic	2, 6, 7, 9, 12, 14, 15, 19, 33, 34, 35, 36, 37, 45, 46, 47, 49, 79, 87, 89, 94, 95, 107, 117, 125
Gumbel Min.	66
Log-Logistic	42, 116, 118
Wakeby	1, 5, 8, 10, 11, 13, 16, 17, 18, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 38, 39, 43, 44, 48, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 61, 62, 63,

	64, 65, 68, 70, 72, 73, 76, 77, 78, 81, 82, 86, 88, 91, 92, 93, 96, 98, 99, 100, 101, 102, 103, 104, 108, 109, 110, 111, 112, 113, 114, 115, 119, 120, 121, 122, 123, 124, 126, 127

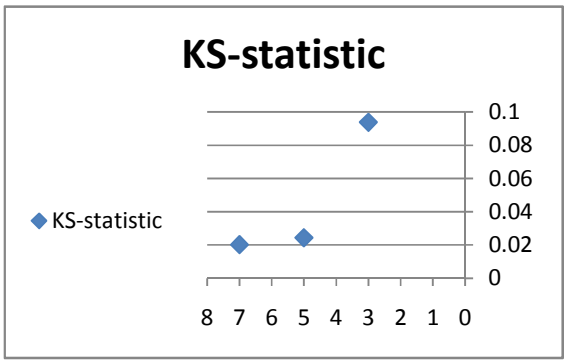


Figure 5.1: average KS-statistic of clustered data

5.4.1 Discussing part 2 of the hypothesis

From Fig 5.1 it is quite obvious that the longer the subsequence the smaller the KS-statistic value. The values of the statistic are 0.0937, 0.0243 and 0.0202 for subsequences of length 3, 5 and 7 respectively. From Fig 5.2 it is apparent that the number of rejected critical values greatly decreases with longer subsequences. For subsequences of length 3 all the five critical points are rejected for all the centroids. Thus subsequences of length 3 have no reliable fit among the tested distributions. 5 residues centroids have better results in terms of the number of rejected points. An average of 2.94 critical points is rejected among all the centroids. Finally centroids of length 7 achieves an average of zero rejected critical point, i.e. all the critical point for all the centroids of length 7 are accepted. Clearly, the length of the subsequence is effective in terms of the KS-statistic and the number of rejected critical values, the thing that proves the second part of the hypothesis.

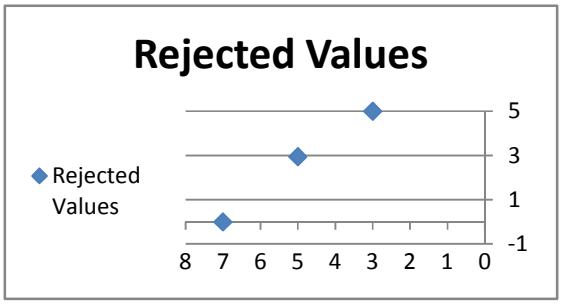


Figure 5.2: number of rejected critical values out of 5 of clustered data

5.4.2 Discussing part 1 of the hypothesis

The same KS-test was performed on unclustered data for n=3, n=5 and n=7. For the three values of n the best fitting distribution was the Wakeby distribution. This is due to the great flexibility of this distribution. The value of the test statistics for the three values of *n* was found to be 0.09041, 0.012 and 0.013 respectively. However these results are not as interesting as they seem to be since all the 5 critical values were rejected for all the values of *n* for unclustered data. The thing that proves the effect of clustering on the final results which in turn emphasis on the existence of a relationship between hydrophobicity pattern and angles measurements, the thing that proves the first part of the hypothesis.

Chapter 6
Conclusion and Future Work

6.1 Conclusion

From the previous chapter it is now clear that there exists a direct relationship between the hydrophobicity of the residues of a subsequence and the measurements of the backbone angles. Classifying a subsequence into one of the available clusters will give a good insight of the angles measurements and consequently the structure of the subsequence. Also the length of the subsequence is an effective factor in angle measurement prediction process. Longer subsequences achieve better fits in one of the standard continuous probability distributions.

6.2 Future work

These results can be used to guide the search process in a complete protein structure prediction algorithm. Using these results can greatly reduce the search space which can increase both the efficiency and the effectiveness of the search process. This angle-hydrophobicity relationship can be used combined with heuristic techniques like genetic algorithm to restrict the initial population to statistically familiar conformation. In this case it is better to apply these guiding rules to only a portion of the initial population in order to leave a chance to the new unfamiliar conformations. Approximations of our results can be applied to crystalline lattices protein models like cube octahedron lattice model which allows the use of several possible angles 60", 90", 120" and 180". Applying the results to this algorithm will allow the predictor to use the most statistically realistic angle of the available alternatives based on its neighboring residues. Also, it is possible to investigate applying the same

approach on subsequences of length more than 7 residues and try to minimize the required processing time.

References

[1] Bonnie Berger, T. L., "Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete." Proceedings of the second annual international conference on Computational molecular biology, vol. 5, issue 1, 1998.

[2] Clayton Matthew Johnson, A. K., "A genetic algorithm with backtracking for protein structure prediction." Proceedings of the 8th annual conference on Genetic and evolutionary computation, 2006.

[3] Diego Rother, G. S., Vijay Pande, "Statistical Characterization of Protein Ensembles." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 5, issue 1, 2008.

[4] Eli Hershkovitz, G. S., Allen Tannenbaum, Loren Dean Williams, "Statistical Analysis of RNA Backbone." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 3, issue 1, 2006.

[5] Eran Segal, D. K., "Probabilistic hierarchical clustering for biological data." Proceedings of the sixth annual international conference on Computational biology, 2002.

[6] Hardik A. Sheth, S. K., "Motif discovery for proteins using subsequence clustering." Proceedings of the 5th international workshop on Bioinformatics, 2005.

[7] Hoos, A. S. a. H. H., "An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem." BMC Bioinformatics, vol. 6, issue 30, 2005.

[8] Jernigan, G. R. a. R. L., "Ideal architecture of residue packing and its observation in protein structures." Protein Science, vol. 6, issue 10, 1997.

[9] Karplus, J. M. C. a. M., "Neural networks for secondary structure and structural class prediction." Protein Science, 1995.

[10] Márcio Dorn, O. N. d. S., "CReF: a central-residue-fragment-based method for predicting approximate 3-D polypeptides structures." Proceedings of the 2008 ACM symposium on applied computing, 2008.

[11] Nancy M. Amato, K. A. D., Guang Song, "Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures." Proceedings of the sixth annual international conference on Computational biology, 2002.

[12] Neal Lesh, M. M., Sue Whitesides, "A complete and effective move set for simplified protein folding." Proceedings of the seventh annual international conference on Research in computational molecular biology, 2003.

[13] Richa Agarwala, S. B., Vlado Dančík, Scott E. Decatur, Martin Farach, Sridhar Hannenhalli, S. Muthukrishnan, Steven Skiena, "Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model." Proceedings of the first annual international conference on Computational molecular biology, 1997.

[14] Rong She, F. C., Ke Wang, Martin Ester, Jennifer L. Gardy and Fiona S. L. Brinkman, "Frequent-subsequence-based prediction of outer membrane proteins." Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003.

[15] Saravanan Dayalan, S. B., Heiko Schroder, "Dihedral angle database of short sub-sequences for protein structure prediction." Proceedings of the second conference on Asia-Pacific bioinformatics 29, 2004

[16] Sergio Raul Duarte Torres, D. C. B. R., Luis Fernando Nino Vasquez, Yoan Jose Pinzon Ardila, "A novel ab-initio genetic-based approach for protein folding prediction." Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007.

[17] Thang N. Bui, G. S., "An efficient genetic algorithm for predicting protein tertiary structures in the 2D HP model." Proceedings of the 2005 conference on Genetic and evolutionary computation, 2005.

[18] Manindra Agrawal, Neeraj Kayal, Nitin Saxena, "PRIMES is in P", Annals of Mathematics 160 (2004), no. 2, pp. 781–793

[19] Andrea Bazzoli, Giorgio Colombo, Andrea G. B. Tettamanzi, "Ab initio protein structure prediction with a dipeptide-assembly evolutionary algorithm", Genetic And Evolutionary Computation Conference (2007)

[20] Altschul, S. F., Gish W., Miller W., Myers E. W., Lipman D. J., "Basic local alignment search tool", (1990), J. Mol. Biol. 215, 403

[21] Gribskov, M., McLachlan, A. D., Eisenberg, D. "Profile analysis: detection of distantly related proteins", (1987), Proc. Natl. Acad. Sci USA. 84, 4355

[22] Krogh A., Brown M., Mian I. S., Sjolander K., Haussler, D. "Hidden Markov models in computational biology: application to protein modeling", (1996), J. Mol. Biol. 235, 1501.

[23] Altschul, S. F., Madden T. L., Schaffer A. A., Zhang J., Zhang Z., Miller W., Lipman D. J. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", (1997), Nucl. Acids Res. 25, 3389.

[24] Karplus K., Barrett C., Hughey R. "Hidden Markov models for detecting remote protein homologies. Bioinformatics", (1998), 14, 846.

[25] Koehl P., Levitt M. "Improved recognition of native-like protein structures using a family of designed sequences", (2002), Proc. Natl. Acad. Sci. 99, 691.

[26] Sauder J. M., Arthur W., and Dunbrack R. L. "Large-scale comparison of protein sequence alignment algorithms with structure alignments", (2000), Proteins: Struct., Func., and Genetics. 40, 6.

[27] Bowie JU, Luthy R, Eisenberg D., "A method to identify protein sequences that fold into a known three-dimensional structure", (1991), Science, 253:164-170.

[28] Jones DT, Taylor WR, Thornton JM, "A new approach to protein fold recognition", (1992), Nature 358:86-89.

[29] Skolnick J, Kihara D, Zhang Y, "Development and large scale benchmark testing of the PROSPECTOR 3.0 threading Algorithm", (2004), Protein, 56:502-518.

[30] Jaroszewski L, Rychlewski L, Li Z, Li W, Godzik A, "FFAS03: a server for profile–profile sequence alignments", (2005), Nucleic Acids Res, 33:W284-W288.

[31] Zhou H, Zhou Y, "Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments", (2005), Proteins, 58:321-328.

[32] Ginalski K, Pas J, Wyrwicz LS, von Grotthuss M, Bujnicki JM, Rychlewski L, "ORFeus: detection of distant homology using sequence profiles and predicted secondary structure", (2003), Nucleic Acids Res, 31:3804-3807.

[33] Shi J, Blundell TL, Mizuguchi K, "FUGUE: sequence–structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties", (2001), JMol Biol, 310:243-257.

[34] Karplus K, Barrett C, Hughey R, "Hidden Markov models for detecting remote protein homologies", (1998), Bioinformatics, 14:846-856.

[35] Soding J, "Protein homology detection by HMM–HMM comparison", (2005), Bioinformatics, 21:951-960.

[36] Jones DT, "GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences", (1999), J Mol Biol, 287:797-815.

[37] Cheng J, Baldi P, "A machine learning information retrieval approach to protein fold recognition", (2006), Bioinformatics, 22:1456-1463.

[38] Kit Fun Lau, Ken A. Dill., "A lattice statistical mechanics model of the conformational and sequence spaces of proteins", (1989), Macromolecules, 22, 3986-3997.

[39] Madhusmita, Harjinder Singh and Abhijit Mitra, "Real Valued Genetic Algorithm Based Approach for Protein Structure Prediction - Role of Biophysical Filters for Reduction of Conformational Search Space", (2008), IEEE EMBC'08, 30th Annual International Conference of the IEEE Engineering in Medicine and Biology, Vancouver, British Columbia, Canada

[40] Xiaolong Zhang, Xiaoli Lin, Chengpeng Wan and Tingting Li, "Genetic-Annealing Algorithm for 3D Off-lattice Protein Folding Model", (2007), Emerging Technologies in Knowledge Discovery and Data Mining, Volume 4819/2007.

[41] Xiaolong Zhang, Wen Cheng, "Protein 3D Structure Prediction by Improved Tabu Search in Off-Lattice AB Model", (2008), Bioinformatics and Biomedical Engineering, Volume, Page(s):184 – 187.

[42] Brocchieri L, Karlin S, "Protein length in eukaryotic and prokaryotic proteomes", (2005), Nucleic Acids Research, 33 (10): 3390-3400

[43] Chothia C., "Proteins. One thousand families for the molecular biologist.", (1992), Nature, 357(6379):543-4

[44] Robert Helling, Hao Li, Régis Mélin, Jonathan Miller, Ned Wingreen, Chen Zeng and Chao Tang, "The designability of protein structures", (2001), Journal of Molecular Graphics and Modeling, Volume 19, Issue 1, Pages 157-167

[45] Wu H., "Studies on Denaturation of Proteins. XIII. A Theory of Denaturation", (1931), Chinese Journal of Physiology 5: 321–344

[46] Anfinsen C. B., "Principles that govern the folding of protein chains", (1973), Science 181 (96): 223–230.

[47] Levinthal, Cyrus, "Are there pathways for protein folding?", (1968), *Journal de Chimie Physique et de Physico-Chimie Biologique* 65: 44–45.

[48] Martin Karplusa and Andrej Šali, "Theoretical studies of protein folding and unfolding", (1995), Current Opinion in Structural Biology Volume 5, Issue 1, Pages 58-73

[49] Dill K.A., "Theory for the folding and stability of globular proteins", (1985), Biochemistry, 24: 1501

[50] Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description Version 3.20 Document Published by the wwPDB

[51] Jeremy M. Berg, John L. Tymoczko, Lubert Stryer, Neil D. Clarke, (2002), "Biochemistry", United States of America: W.H. Freeman

Appendix (I)

Hydrophobicity values of the final k-means centroids

Table 1: Final centroids of k-means clustering of subsequences of length 3

$n = 3$		
$C_0 = (3.26, 3.27, 3.22)$	$C_1 = (3.28, 3.29, -2.49)$	$C_2 = (3.27, -2.43, 3.25)$
$C_3 = (3.34, -2.46, -2.47)$	$C_4 = (-2.54, 3.28, 3.32)$	$C_5 = (-2.39, 3.34, -2.42)$
$C_6 = (-2.48, -2.45, 3.35)$	$C_7 = (-2.35, -2.37, -2.36)$	

Table 2: Final centroids of k-means clustering of subsequences of length

$n = 5$	
$C_0 = (3.13, 3.23, 3.17, 3.05, 3.07)$	$C_1 = (3.33, 3.22, 3.19, 3.21, -2.36)$
$C_2 = (3.22, 3.25, 3.14, -2.54, 3.11)$	$C_3 = (3.28, 3.33, 3.28, -2.42, -2.42)$
$C_4 = (3.15, 3.22, -2.67, 3.08, 3.23)$	$C_5 = (3.31, 3.24, -2.41, 3.17, -2.37)$
$C_6 = (3.23, 3.28, -2.61, -2.57, 3.25)$	$C_7 = (3.34, 3.35, -2.41, -2.48, -2.36)$
$C_8 = (3.16, -2.63, 3.21, 3.29, 3.20)$	$C_9 = (3.29, -2.53, 3.18, 3.28, -2.51)$
$C_{10} = (3.25, -2.33, 3.25, -2.31, 3.29)$	$C_{11} = (3.29, -2.36, 3.32, -2.40, -2.41)$
$C_{12} = (3.29, -2.59, -2.62, 3.27, 3.30)$	$C_{13} = (3.32, -2.54, -2.44, 3.33, -2.45)$
$C_{14} = (3.33, -2.37, -2.46, -2.42, 3.35)$	$C_{15} = (3.38, -2.40, -2.43, -2.39, -2.34)$
$C_{16} = (-2.59, 3.29, 3.24, 3.25, 3.19)$	$C_{17} = (-2.53, 3.28, 3.36, 3.25, -2.51)$
$C_{18} = (-2.54, 3.25, 3.27, -2.50, 3.15)$	$C_{19} = (-2.53, 3.28, 3.35, -2.52, -2.57)$
$C_{20} = (-2.37, 3.27, -2.50, 3.25, 3.32)$	$C_{21} = (-2.40, 3.31, -2.32, 3.36, -2.37)$
$C_{22} = (-2.38, 3.34, -2.53, -2.47, 3.34)$	$C_{23} = (-2.38, 3.37, -2.38, -2.42, -2.44)$

$C_{24} = (-2.41, -2.50, 3.33, 3.33, 3.27)$	$C_{25} = (-2.52, -2.53, 3.33, 3.34, -2.51)$
$C_{26} = (-2.52, -2.42, 3.32, -2.42, 3.33)$	$C_{27} = (-2.43, -2.40, 3.39, -2.47, -2.46)$
$C_{28} = (-2.37, -2.41, -2.46, 3.37, 3.36)$	$C_{29} = (-2.39, -2.42, -2.39, 3.38, -2.45)$
$C_{30} = (-2.36, -2.39, -2.39, -2.41, 3.39)$	$C_{31} = (-2.31, -2.28, -2.26, -2.27, -2.32)$

Table 3: Final centroids of k-means clustering of subsequences of length 7

$n = 7$	
$C_0 = (3.10, 3.20, 3.07, 3.19, 2.98, 2.82, 2.90)$	$C_1 = (3.20, 3.12, 3.21, 3.22, 2.92, 3.02, -2.52)$
$C_2 = (3.12, 3.22, 3.27, 2.79, 3.01, -2.62, 2.96)$	$C_3 = (3.09, 3.28, 3.11, 3.06, 3.17, -2.45, -2.62)$
$C_4 = (3.16, 3.04, 3.12, 3.19, -2.74, 3.12, 3.17)$	$C_5 = (3.32, 3.24, 3.09, 3.14, -2.37, 3.00, -2.17)$
$C_6 = (3.23, 3.13, 3.25, 3.15, -2.56, -2.45, 3.15)$	$C_7 = (3.44, 3.29, 3.23, 3.28, -2.19, -2.41, -2.26)$
$C_8 = (3.18, 2.95, 3.14, -2.66, 2.98, 3.14, 3.02)$	$C_9 = (3.17, 3.12, 3.19, -2.72, 3.10, 3.27, -2.49)$
$C_{10} = (3.22, 3.26, 3.16, -2.39, 3.02, -2.10, 3.19)$	$C_{11} = (3.25, 3.40, 3.10, -2.46, 3.19, -2.41, -2.34)$
$C_{12} = (3.13, 3.21, 3.25, -2.56, -2.60, 3.23, 3.17)$	$C_{13} = (3.22, 3.34, 3.21, -2.60, -2.34, 3.21, -2.39)$
$C_{14} = (3.27, 3.35, 3.27, -2.26, -2.39, -2.26, 3.30)$	$C_{15} = (3.37, 3.34, 3.33, -2.37, -2.42, -2.26, -2.25)$
$C_{16} = (3.04, 3.11, -2.52, 2.93, 3.19, 3.12, 2.94)$	$C_{17} = (3.05, 3.20, -2.68, 3.06, 3.21, 3.11, -2.67)$
$C_{18} = (3.13, 3.28, -2.62, 3.05, 3.06, -2.65, 3.00)$	$C_{19} = (3.22, 3.24, -2.72, 3.12, 3.31, -2.59, -2.68)$
$C_{20} = (3.14, 3.13, -2.09, 3.01, -2.30, 3.11, 3.18)$	$C_{21} = (3.31, 3.25, -2.36, 3.16, -2.30, 3.28, -2.37)$
$C_{22} = (3.39, 3.31, -2.48, 3.18, -2.54, -2.54, 3.18)$	$C_{23} = (3.30, 3.21, -2.49, 3.23, -2.32, -2.33, -2.46)$
$C_{24} = (3.04, 3.27, -2.62, -2.54, 3.11, 3.15, 3.08)$	$C_{25} = (3.16, 3.28, -2.59, -2.75, 3.24, 3.29, -2.63)$
$C_{26} = (3.26, 3.19, -2.63, -2.57, 3.17, -2.37, 3.28)$	$C_{27} = (3.29, 3.32, -2.59, -2.46, 3.32, -2.48, -2.51)$
$C_{28} = (3.34, 3.26, -2.27, -2.59, -2.33, 3.32, 3.28)$	$C_{29} = (3.27, 3.35, -2.39, -2.46, -2.30, 3.37, -2.39)$
$C_{30} = (3.32, 3.35, -2.43, -2.56, -2.36, -2.43, 3.35)$	$C_{31} = (3.38, 3.38, -2.43, -2.39, -2.39, -2.30, -2.32)$
$C_{32} = (3.06, -2.44, 2.98, 3.28, 3.17, 3.02, 3.08)$	$C_{33} = (3.05, -2.66, 3.30, 3.18, 3.28, 3.16, -2.28)$

$C_{34} = (3.18, -2.67, 3.20, 3.25, 3.08, -2.47, 3.08)$	$C_{35} = (3.22, -2.63, 3.22, 3.38, 3.24, -2.66, -2.51)$
$C_{36} = (3.31, -2.45, 3.03, 3.15, -2.60, 3.10, 3.15)$	$C_{37} = (3.33, -2.44, 3.09, 3.28, -2.33, 3.16, -2.47)$
$C_{38} = (3.18, -2.61, 3.10, 3.22, -2.59, -2.65, 3.20)$	$C_{39} = (3.34, -2.55, 3.31, 3.36, -2.54, -2.56, -2.41)$
$C_{40} = (3.06, -2.20, 3.10, -2.42, 3.11, 3.29, 3.15)$	$C_{41} = (3.24, -2.34, 3.25, -2.32, 3.22, 3.23, -2.48)$
$C_{42} = (3.23, -2.30, 3.26, -2.21, 3.29, -2.46, 3.29)$	$C_{43} = (3.32, -2.37, 3.28, -2.33, 3.37, -2.41, -2.46)$
$C_{44} = (3.23, -2.25, 3.21, -2.51, -2.66, 3.33, 3.31)$	$C_{45} = (3.28, -2.35, 3.30, -2.50, -2.42, 3.25, -2.39)$
$C_{46} = (3.21, -2.44, 3.30, -2.28, -2.33, -2.47, 3.24)$	$C_{47} = (3.36, -2.37, 3.37, -2.35, -2.32, -2.40, -2.35)$
$C_{48} = (3.22, -2.57, -2.68, 3.27, 3.16, 3.17, 3.07)$	$C_{49} = (3.32, -2.41, -2.46, 3.26, 3.32, 3.13, -2.52)$
$C_{50} = (3.24, -2.66, -2.76, 3.30, 3.25, -2.62, 3.04)$	$C_{51} = (3.32, -2.62, -2.60, 3.25, 3.35, -2.53, -2.64)$
$C_{52} = (3.19, -2.55, -2.45, 3.24, -2.52, 3.26, 3.30)$	$C_{53} = (3.33, -2.59, -2.48, 3.24, -2.33, 3.34, -2.47)$
$C_{54} = (3.34, -2.44, -2.43, 3.40, -2.53, -2.45, 3.42)$	$C_{55} = (3.35, -2.56, -2.41, 3.39, -2.45, -2.59, -2.48)$
$C_{56} = (3.26, -2.46, -2.50, -2.38, 3.33, 3.34, 3.27)$	$C_{57} = (3.28, -2.26, -2.55, -2.45, 3.32, 3.27, -2.48)$
$C_{58} = (3.31, -2.37, -2.52, -2.39, 3.35, -2.54, 3.29)$	$C_{59} = (3.39, -2.40, -2.37, -2.44, 3.37, -2.45, -2.47)$
$C_{60} = (3.40, -2.48, -2.56, -2.38, -2.49, 3.35, 3.48)$	$C_{61} = (3.34, -2.30, -2.46, -2.47, -2.39, 3.39, -2.46)$
$C_{62} = (3.36, -2.41, -2.42, -2.38, -2.29, -2.39, 3.45)$	$C_{63} = (3.41, -2.42, -2.34, -2.35, -2.28, -2.36, -2.44)$
$C_{64} = (-2.44, 3.17, 3.16, 3.18, 3.28, 2.97, 3.00)$	$C_{65} = (-2.45, 3.10, 3.28, 3.13, 3.00, 3.18, -2.48)$
$C_{66} = (-2.69, 3.29, 3.13, 3.25, 3.21, -2.42, 3.08)$	$C_{67} = (-2.64, 3.41, 3.29, 3.32, 3.26, -2.25, -2.33)$
$C_{68} = (-2.52, 3.24, 3.05, 3.18, -2.70, 3.03, 3.27)$	$C_{69} = (-2.53, 3.24, 3.47, 3.12, -2.46, 3.19, -2.37)$
$C_{70} = (-2.61, 3.22, 3.33, 3.26, -2.61, -2.41, 3.25)$	$C_{71} = (-2.47, 3.36, 3.41, 3.33, -2.42, -2.41, -2.27)$
$C_{72} = (-2.50, 3.11, 3.17, -2.63, 3.04, 3.22, 3.16)$	$C_{73} = (-2.64, 3.24, 3.26, -2.67, 3.10, 3.22, -2.67)$
$C_{74} = (-2.52, 3.23, 3.22, -2.19, 3.15, -2.40, 3.23)$	$C_{75} = (-2.49, 3.31, 3.33, -2.51, 3.22, -2.40, -2.46)$
$C_{76} = (-2.58, 3.10, 3.29, -2.62, -2.74, 3.20, 3.28)$	$C_{77} = (-2.54, 3.25, 3.31, -2.61, -2.58, 3.30, -2.47)$
$C_{78} = (-2.55, 3.27, 3.35, -2.41, -2.57, -2.34, 3.38)$	$C_{79} = (-2.47, 3.37, 3.39, -2.47, -2.49, -2.45, -2.41)$
$C_{80} = (-2.38, 3.06, -2.61, 3.32, 3.20, 3.29, 3.17)$	$C_{81} = (-2.40, 3.22, -2.62, 3.29, 3.40, 3.22, -2.56)$

$C_{82} = (-2.35, 3.35, -2.35, 3.08, 3.33, -2.30, 3.23)$	$C_{83} = (-2.36, 3.32, -2.47, 3.31, 3.31, -2.53, -2.53)$
$C_{84} = (-2.35, 3.23, -2.42, 3.31, -2.40, 3.23, 3.28)$	$C_{85} = (-2.51, 3.32, -2.33, 3.33, -2.29, 3.38, -2.45)$
$C_{86} = (-2.26, 3.25, -2.24, 3.32, -2.49, -2.50, 3.33)$	$C_{87} = (-2.46, 3.37, -2.33, 3.43, -2.33, -2.32, -2.39)$
$C_{88} = (-2.44, 3.32, -2.38, -2.52, 3.33, 3.32, 3.16)$	$C_{89} = (-2.36, 3.31, -2.66, -2.61, 3.28, 3.33, -2.53)$
$C_{90} = (-2.38, 3.34, -2.55, -2.42, 3.28, -2.41, 3.33)$	$C_{91} = (-2.38, 3.37, -2.46, -2.39, 3.43, -2.49, -2.55)$
$C_{92} = (-2.32, 3.28, -2.36, -2.50, -2.49, 3.32, 3.30)$	$C_{93} = (-2.48, 3.38, -2.39, -2.40, -2.49, 3.35, -2.55)$
$C_{94} = (-2.38, 3.38, -2.34, -2.46, -2.48, -2.42, 3.39)$	$C_{95} = (-2.35, 3.40, -2.40, -2.36, -2.35, -2.27, -2.42)$
$C_{96} = (-2.40, -2.45, 3.17, 3.24, 3.14, 3.11, 3.12)$	$C_{97} = (-2.39, -2.65, 3.40, 3.27, 3.31, 3.28, -2.32)$
$C_{98} = (-2.44, -2.46, 3.26, 3.36, 3.18, -2.61, 3.16)$	$C_{99} = (-2.42, -2.45, 3.38, 3.38, 3.34, -2.38, -2.35)$
$C_{100} = (-2.61, -2.66, 3.28, 3.28, -2.69, 3.07, 3.25)$	$C_{101} = (-2.43, -2.52, 3.39, 3.30, -2.43, 3.21, -2.37)$
$C_{102} = (-2.57, -2.52, 3.25, 3.35, -2.63, -2.64, 3.30)$	$C_{103} = (-2.48, -2.48, 3.36, 3.38, -2.39, -2.49, -2.41)$
$C_{104} = (-2.44, -2.47, 3.19, -2.69, 3.37, 3.35, 3.27)$	$C_{105} = (-2.54, -2.37, 3.38, -2.48, 3.23, 3.36, -2.43)$
$C_{106} = (-2.58, -2.52, 3.31, -2.44, 3.34, -2.26, 3.34)$	$C_{107} = (-2.51, -2.38, 3.34, -2.27, 3.39, -2.39, -2.36)$
$C_{108} = (-2.60, -2.46, 3.37, -2.61, -2.54, 3.28, 3.34)$	$C_{109} = (-2.35, -2.40, 3.39, -2.49, -2.39, 3.45, -2.50)$
$C_{110} = (-2.54, -2.41, 3.36, -2.43, -2.50, -2.50, 3.40)$	$C_{111} = (-2.33, -2.36, 3.41, -2.40, -2.46, -2.41, -2.32)$
$C_{112} = (-2.50, -2.32, -2.54, 3.34, 3.30, 3.28, 3.28)$	$C_{113} = (-2.37, -2.43, -2.46, 3.37, 3.41, 3.38, -2.43)$
$C_{114} = (-2.30, -2.43, -2.47, 3.37, 3.31, -2.49, 3.22)$	$C_{115} = (-2.37, -2.43, -2.40, 3.37, 3.39, -2.45, -2.47)$
$C_{116} = (-2.42, -2.47, -2.39, 3.33, -2.60, 3.29, 3.39)$	$C_{117} = (-2.40, -2.50, -2.39, 3.39, -2.33, 3.39, -2.24)$
$C_{118} = (-2.49, -2.46, -2.43, 3.38, -2.54, -2.45, 3.35)$	$C_{119} = (-2.28, -2.31, -2.36, 3.40, -2.39, -2.39, -2.43)$
$C_{120} = (-2.47, -2.41, -2.32, -2.54, 3.37, 3.38, 3.36)$	$C_{121} = (-2.38, -2.39, -2.36, -2.43, 3.40, 3.40, -2.46)$
$C_{122} = (-2.36, -2.43, -2.47, -2.39, 3.38, -2.38, 3.38)$	$C_{123} = (-2.28, -2.36, -2.39, -2.35, 3.40, -2.46, -2.38)$
$C_{124} = (-2.45, -2.37, -2.29, -2.32, -2.46, 3.42, 3.33)$	$C_{125} = (-2.36, -2.33, -2.33, -2.39, -2.35, 3.39, -2.41)$
$C_{126} = (-2.32, -2.39, -2.29, -2.27, -2.41, -2.40, 3.37)$	$C_{127} = (-2.20, -2.14, -2.18, -2.14, -2.14, -2.18, -2.16)$

Appendix (II)

NP-Completeness

In computational complexity theory, **P**, also known as PTIME, is one of the most fundamental complexity classes. It contains all decision problems which can be solved by a deterministic Turing machine using a polynomial amount of computation time, or polynomial time.

Cobham's thesis holds that P is the class of computational problems which are "efficiently solvable" or "tractable"; in practice, some problems not known to be in P have practical solutions, and some that are in P do not, but this is a useful rule of thumb.

P is known to contain many natural problems, including the decision versions of linear programming, calculating the greatest common divisor, and finding a maximum matching. In 2002, it was shown that the problem of determining if a number is prime is in P [18].

In computational complexity theory, NP is one of the most fundamental complexity classes. The abbreviation NP refers to "Nondeterministic Polynomial time".

Intuitively, NP is the set of all decision problems for which the 'yes'-answers have simple proofs of the fact that the answer is indeed 'yes'. More precisely, these proofs have to be verifiable in polynomial time by a deterministic Turing machine. In an equivalent formal definition, NP is the set of decision problems solvable in polynomial time by a non-deterministic Turing machine.

NP-complete is a subset of NP, the set of all decision problems whose solutions can be verified in polynomial time; NP may be equivalently defined as the set of decision problems that can be solved in polynomial time on a nondeterministic Turing machine. A problem p in NP is also in NPC if and only if every other problem in NP can be transformed into p in polynomial time. Subset sum problem is a famous example of NP-Complete problems.

Subset sum problem

The problem is this: given a set of integers, does the sum of some non-empty subset equal exactly zero? For example, given the set {−7, −3, −2, 5, 8}, the answer is YES because the subset {−3, −2, 5} sums to zero.

Other well-known NP-complete problems

- Boolean satisfiability problem (SAT)
- N-puzzle
- Knapsack problem
- Hamiltonian path problem
- Travelling salesman problem
- Subgraph isomorphism problem
- Clique problem
- Vertex cover problem
- Independent set problem
- Dominating set problem
- Graph coloring problem

In 1998 Bonnie Berger and Tom Leighton have shown that protein folding in the HP model is NP-complete. They have thus "settled the recurring open question about the complexity of protein folding in this

model" in their own terms. Their work also complements existing efforts to characterize various hardness aspects of protein folding. In particular, their proof shows that any reasonably fast protein folding algorithm will have to rely on aspects other than just hydrophobicity considerations. However their methods do not apply to the 2D square lattice, although the 3D cubic model that they studied seems to be more relevant to the actual protein folding problem. Nor does their proof directly apply to the 3D tetrahedral lattice.

Appendix (III)

Constraint Satisfaction Problem

What is a constraint satisfaction problem?

This is a family of problems not a single problem. A problem from this family deals with constraints. There are constraints all around us, such as managing work and home life and making sure we don't go over budget, and we figure out ways to deal with them to varying success. Sometimes we fail and this is most often due to our limited capacity to deal with problems involving a large amount of variables and constraints. This is where computers, and more specifically, constraint satisfaction problems (CSPs), are necessary.

Like most problems in artificial intelligence (AI), CSPs are solved through search. CSPs – unlike other AI problems – have a standard structure that allows general search algorithms using heuristics to be implemented for any CSP. The structure of the CSP problem is largely independent of its domain. These special and defining characteristics make CSPs both interesting and worthwhile to study.

What are the practical applications of CSPs?

CSPs best suites general temporal and combinatorial problems, among other things. The following are examples where constraint programming has been successfully applied:

1. Operations Research (scheduling, timetabling)
2. Bioinformatics (DNA sequencing, protein folding)
3. Electrical engineering (circuit layout-ing)
4. Telecommunications
5. Hubbell telescope/Satellite scheduling

Generally speaking, CSPs are a rather recent formulation. There is not extensive published literature on the subject, but they are widely studied and their applications will continue to increase.

Definition of a CSP

Formal definition

The formal definition of a CSP involves variables and their domains, and constraints. Suppose we have a set of variables, X_1, X_2... X_n, all with domains D_1, D_2... D_n such that all variables X_i have a value in their respective domain D_i. There is also a set of constraints, C_1, C_2... C_m, such that a constraint C_i restricts (imposes a constraint on) the possible values in the domain of some subset of the variables. A solution to a CSP is an assignment of every variable some value in its domain such that every constraint is satisfied. Therefore, each assignment (a state change or step in a search) of a value to a variable must be consistent: it must not violate any of the constraints.

As in any AI search problem, there can be multiple solutions (or none). To address this, a CSP may have a preference of one solution over another using some preference.

Finite vs. real-valued domains

Here we are concerned only with CSP's that have finite domain variables. This means that the domains are a finite set of integers, as opposed to a real-valued domain that would include an infinite number of real-values between two bounds.

The modeling of a real problem

Consider the popular N-Queens problem used throughout AI. This problem involves placing n queens on an n x n chessboard such that no queen is attacking another queen. (According to the rules of chess, a queen is able to attack another piece – in this case, a queen – if it is in the

same row, column, or diagonal from that queen.) There are, of course, many ways to formulate this problem as a CSP (think: variables, domains, and constraints). A simple model is to represent each queen as a row so that (for example) to solve the 4-queen problem, we have variables Q_1, Q_2, Q_3, and Q_4. Each of these variables has an integer domain, whose values correspond to the different columns, 1-4. An assignment consists of assigning a column to a queen, i.e. $\{Q_1 = 2\}$, which "places" a queen in row 1, column 2. The constraints on the problem restrict certain values for each variable so that all assignments are consistent. For example, after we have assigned Q_1, and now want to assign Q_2, we know we cannot use value 2, since this would violate a constraint: Q_1 could attack Q_2 and vice versa. Thus we come up with the following variables, values, and constraints to model this problem:

Variables: { Q_1, Q_2, Q_3, Q_4 }
Domain: { (1, 2, 3, 4), (1, 2, 3, 4), (1, 2, 3, 4), (1, 2, 3, 4) }
Constraints: *AllDifferent*(Q_1, Q_2, Q_3, Q_4) and
for i = 0...n and j = (i+1)...n, k = j-i, Q[i] != Q[j] + k and Q[i] != Q[j] - k.