3-Dimensional Protein Reconstruction from Contact Maps: Complexity and Experimental Results

Filippo Medri

Technical Report UBLCS-2009-08

March 2009

Department of Computer Science

University of Bologna

Mura Anteo Zamboni 7
40127 Bologna (Italy)

The University of Bologna Department of Computer Science Research Technical Reports are available in PDF and gzipped PostScript formats via anonymous FTP from the area `ftp.cs.unibo.it:/pub/TR/UBLCS` or via WWW at URL `http://www.cs.unibo.it/`. Plain-text abstracts organized by year are available in the directory `ABSTRACTS`.

Recent Titles from the UBLCS Technical Report Series

2006-22 *Broadcasting at the Critical Threshold*, Arteconi, S., Hales, D., October 2006.

2006-23 *Emergent Social Rationality in a Peer-to-Peer System*, Marcozzi, A., Hales, D., October 2006.

2006-24 *Reconstruction of the Protein Structures from Contact Maps*, Margara, L., Vassura, M., di Lena, P., Medri, F., Fariselli, P., Casadio, R., October 2006.

2006-25 *Lambda Types on the Lambda Calculus with Abbreviations*, Guidi, F., November 2006.

2006-26 *FirmNet: The Scope of Firms and the Allocation of Task in a Knowledge-Based Economy*, Mollona, E., Marcozzi, A. November 2006.

2006-27 *Behavioral Coalition Structure Generation*, Rossi, G., November 2006.

2006-28 *On the Solution of Cooperative Games*, Rossi, G., December 2006.

2006-29 *Motifs in Evolving Cooperative Networks Look Like Protein Structure Networks*, Hales, D., Arteconi, S., December 2006.

2007-01 *Extending the Choquet Integral*, Rossi, G., January 2007.

2007-02 *Towards Cooperative, Self-Organised Replica Management*, Hales, D., Marcozzi, A., Cortese, G., February 2007.

2007-03 *A Model and an Algebra for Semi-Structured and Full-Text Queries (PhD Thesis)*, Buratti, G., March 2007.

2007-04 *Data and Behavioral Contracts for Web Services (PhD Thesis)*, Carpineti, S., March 2007.

2007-05 *Pattern-Based Segmentation of Digital Documents: Model and Implementation (PhD Thesis)*, Di Iorio, A., March 2007.

2007-06 *A Communication Infrastructure to Support Knowledge Level Agents on the Web (PhD Thesis)*, Guidi, D., March 2007.

2007-07 *Formalizing Languages for Service Oriented Computing (PhD Thesis)*, Guidi, C., March 2007.

2007-08 *Secure Gossiping Techniques and Components (PhD Thesis)*, Jesi, G., March 2007.

2007-09 *Rich Media Content Adaptation in E-Learning Systems (PhD Thesis)*, Mirri, S., March 2007.

2007-10 *User Interaction Widgets for Interactive Theorem Proving (PhD Thesis)*, Zacchiroli, S., March 2007.

2007-11 *An Ontology-based Approach to Define and Manage B2B Interoperability (PhD Thesis)*, Gessa, N., March 2007.

2007-12 *Decidable and Computational Properties of Cellular Automata (PhD Thesis)*, Di Lena, P., March 2007.

Dottorato di Ricerca in Informatica

Università di Bologna e Padova

3-Dimensional Protein Reconstruction from Contact Maps: Complexity and Experimental Results

Filippo Medri

March 2009

Coordinatore:	Tutore:
Simone Martini	Luciano Margara

Abstract

The prediction of the protein tertiary structure from solely its residue sequence is one of the most challenging problems in structural bioinformatics. Predicting the tertiary structure of a protein directly from its primary structure is a complex problem. A typical alternative approach is to identify a set of sub-problems, such as prediction of residue contacts and try to reconstruct the three-dimensional structure from this partial information. The general problem of recovering a set of three-dimensional coordinates consistent with some given contact map is known as unit disk graph realization problem and it's recently proven to be NP-hard. The specific protein reconstruction problem poses further constraints to the general realization problem. In particular, proteins always presents a typical ordered substructure which is called backbone. In the first part of this thesis we investigate the computational complexity of the protein reconstruction problem and prove that the 2-dimensional realization problem remains NP-hard even with the backbone constraint. In the second part of the thesis we present COMAR, an heuristic algorithm for the reconstruction of protein 3D-structure from contact map. Such algorithm has been tested on a non redundant data set consisting of 1760 proteins. and it was always able to produce three-dimensional coordinates consistent with the initial contact map for the whole data set. Performance analysis of the algorithm shows that there exist native contact maps for which there are numerous different possible structures consistent with them. We proceed further to evaluate the fault tolerance of COMAR introducing three different class of random errors. The analysis shows that

the algorithm tolerates error on contact. We introduce then an improved version of the algorithm, called FT-COMAR (fault tolerant COMAR), which experimental results show that it can ignore up to 75% of the contact map and still obtain a protein three-dimensional structures whose RMSD for the native one is less then 4 Armstrong. Furthermore the reconstruction quality is independent from protein length, which suggest that, to improve protein reconstruction from contact maps, contact map prediction should put more emphasis on prediction quality instead of quantity.

Contents

List of Tables

List of Figures

0.1 Introduction

We devoted a large part of our PHD designing and developing COMAR [16, 15] (Contact Map Reconstraction) a heuristic method that is able to reconstruct with an unprecedented rate (3-15 seconds) a 3D model that exactly matches the target contact map of a protein. The heuristic was born in the computer science department of the University of Bologna as a research work of the Bioinformatics group (to which we belong) with the partnership of the Bologna BioComputing group. The motivations behind the development of COMAR were the following:

- Analyzing the computational aspects of the protein reconstruction problem from a distance geometry point of view with the aim of understanding the source of the complexity behind protein folding.
- Synthesizing an efficient heuristic method able to tackle distance geometry description not relying on peculiar features of some given sample set.
- Experimenting new solutions to exploit the results coming from contact map predictions.

These motivations comes from a series of considerations. First of all contact map prediction has proved itself to be very successful and continue to offer several interesting challenges with regards to methods and analysis (see [13, 20]). On the other hand the problem of protein reconstruction from contact maps was faced in literature several times, while every time with a very narrow scope (see [25, 18, 17]), failing to tackle the problem from a general point of view. Besides committing ourselves in the design and development of COMAR we work on proving that the problem of the realization of a graph in a K-dimensional space with the constraint of a backbone of equidistant points is NP-hard. The proof is realized by a polynomial time reduction to the 3-SAT problem ([19]), inspired by the work of Breu and Kirkpatrick ([9]).

Therefore our PHD was devoted to studying the problem of reconstruction of a k-dimensional structure starting from the information given by a contact map with the constraint of a backbone. A contact map of a given protein P is a binary matrix M such that $M_{i,j} = 1$ if and only if the physical distance between residues i and j in the native structure is less than or equal to a pre-assigned threshold t. The contact map of each protein is a distinctive signature of its folded structure. Predicting the tertiary structure of a protein directly from its primary structure is a very complex and still unsolved problem. An alternative and probably more feasible approach is to predict the contact map of a protein from its primary structure and then to compute the tertiary structure starting from the predicted contact map, which is the problem we have shown to be NP-Hard. The thesis is composed of two parts where we respectively describe:

- The theoretical results by which we have proved that the realization of a graph in a k-dimensional space with a backbone of equidistant points constraint is NP-Hard.

- The design of COMAR, its performance and the results of its application.

In the first part we introduce the protein folding problem and the major results concerning the prediction of protein structures. We present then problem of the realization of a a graph in a k-dimensional space from a distance geometry point of view. We then introduce the backbone constraint and then we show a polynomial time reduction of the problem to the 3-SAT problem.

In the second part we present the COMAR [16] heuristic, its performance and an update of the algorithm (FT-COMAR [Fault Tolerant-COMAR] [15]) able to reconstruct the three-dimensional structure of a protein starting from faulty contact maps. COMAR computes an exact model for the protein independently of the contact map threshold with 100% efficiency when tested on 1760 proteins from different structural classes. Repeated applications of COMAR (starting from randomly chosen distinct initial solutions) show that the same contact map may admit (depending on the threshold) quite different 3D models. Extensive experimental

results show that contact map thresholds ranging from 10 to 18 Ångstrom allow reconstructing 3D models that are very similar to the proteins native structure. In order to simulate possible scenarios of reconstruction from predicted (and therefore highly noised) contact maps we test the performances of COMAR on native contact maps when a perturbation with random errors is introduced. From our analysis we obtain that our algorithm performs better reconstructions on blurred contact maps when contacts are under predicted than over predicted. In chapter 5 we give a new version of the algorithm which can be used with incomplete contact maps. FT-COMAR can ignore up to 75% of the contact map and still recover from the remaining 25% entries a three dimensional structure whose root mean square deviation (RMSD) from the native one is less then 4 Å. Our results indicate that the quality more than the quantity of predicted contacts is relevant to the protein 3D reconstruction and that some hints about unsafe areas in the predicted contact maps can be useful to improve reconstruction quality. For this, we implement a very simple filtering procedure to detect unsafe areas in contact maps and we show that by this and in the presences of errors the performance of the algorithm can be significantly improved. Furthermore, we show that both COMAR and FT-COMAR overcome previous state-of-the-art algorithms for the same task. Finally in the 5.8 we show how we have realized cages of capacity 2 with regard to bead configurations.

Part I

Theoretical Results

Chapter 1

Structural BioInformatics

Proteins constitutes a macromolecular class of enormous importance from a biological point of view. Like other biological macromolecules such as polysaccharides and nucleic acids, proteins are essential parts of organisms and participate in every process within cells. Many proteins are enzymes that catalyze biochemical reactions, and are vital to metabolism. Proteins also have structural or mechanical functions, such as actin and myosin in muscle, and the proteins in the cytoskeleton, which forms a system of scaffolding that maintains cell shape. Other proteins are important in cell signaling, immune responses, cell adhesion, and the cell cycle. Protein is also necessary in animals' diets, since they cannot synthesize all the amino acids and must obtain essential amino acids from food. Through the process of digestion, animals break down ingested protein into free amino acids that are then used in metabolism.

From a chemical point of view proteins are complex eteropolymer, they are chains of 20 fundamental subunits, called amino-acids, made up from 50 to 1000 units.

1.1 Protein Structure

Biochemistry refers to four distinct aspects of a protein's structure:

- Primary structure - the amino acid sequence of the peptide chains.

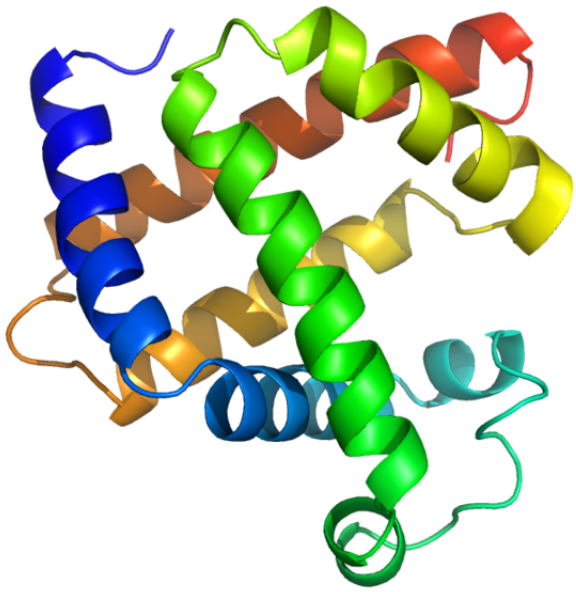


Figure 1.1: Myoglobin

- Secondary structure - highly regular sub-structures (alpha helix and strands of beta sheet) which are locally defined, meaning that there can be many different secondary motifs present in one single protein molecule.

- Tertiary structure - Three-dimensional structure of a single protein molecule; a spatial arrangement of the secondary structures.

- Quaternary structure - complex of several protein molecules or polypeptide chains, usually called protein subunits in this context, which function as part of the larger assembly or protein complex.

In addition to these levels of structure, a protein may shift between several similar structures in performing its biological function. In the context of these functional rearrangements, these tertiary or quaternary structures are usually referred to as chemical conformation, and transitions between them are called conformational changes.

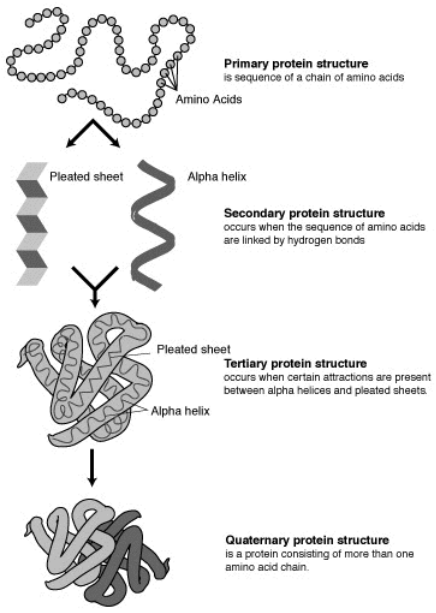


Figure 1.2: Protein structures

1.1.1 Aminoacids

The primary structure is held together by covalent or peptide bonds, which are made during the process of protein biosynthesis or translation. These peptide bonds provide rigidity to the protein. The two ends of the amino acid chain are referred to as the C-terminal end or carboxyl terminus (C-terminus) and the N-terminal end or amino terminus (N-terminus) based on the nature of the free group on each extremity.

The various types of secondary structure are defined by their patterns of hydrogen bonds between the main-chain peptide groups. However, these hydrogen bonds are generally not stable by themselves, since the water-amide hydrogen bond is generally more favorable than the amide-amide hydrogen bond. Thus, secondary structure is stable only when the local concentration of water is sufficiently low, e.g., in the molten globule or fully folded states.

Similarly, the formation of molten globules and tertiary structure is driven mainly

by structurally non-specific interactions, such as the rough propensities of the amino acids and hydrophobic interactions. However, the tertiary structure is fixed only when the parts of a protein domain are locked into place by structurally specific interactions, such as ionic interactions (salt bridges), hydrogen bonds and the tight packing of side chains. The tertiary structure of extracellular proteins can also be stabilized by disulfide bonds, which reduce the entropy of the unfolded state; disulfide bonds are extremely rare in cytosolic proteins, since the cytosol is generally a reducing environment.

An α-amino acid consists of a part that is present in all the amino acid types, and a side chain that is unique to each type of residue. The C_α atom is bound to 4 different molecules (the H is omitted in the diagram); an amino group, a carboxyl group, a hydrogen and a side chain, specific for this type of amino acid. An exception from this rule is proline, where the hydrogen atom is replaced by a bond to the side chain. Because the carbon atom is bound to four different groups it is chiral, however only one of the isomers occur in biological proteins. Glycine however, is not chiral since its side chain is a hydrogen atom. A simple mnemonic for correct L-form is "CORN": when the C_α atom is viewed with the H in front, the residues read "CO-R-N" in a clockwise direction.

The side chain determines the chemical properties of the α-amino acid and may be any one of the 20 different side chains:

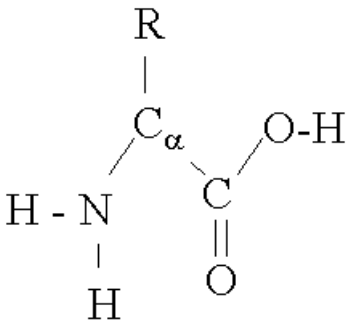


Figure 1.3: α-amino acid

1.1.2 Protein Folding

Protein folding is the physical process by which a polypeptide folds into its characteristic three-dimensional structure. Each protein begins as a polypeptide, translated

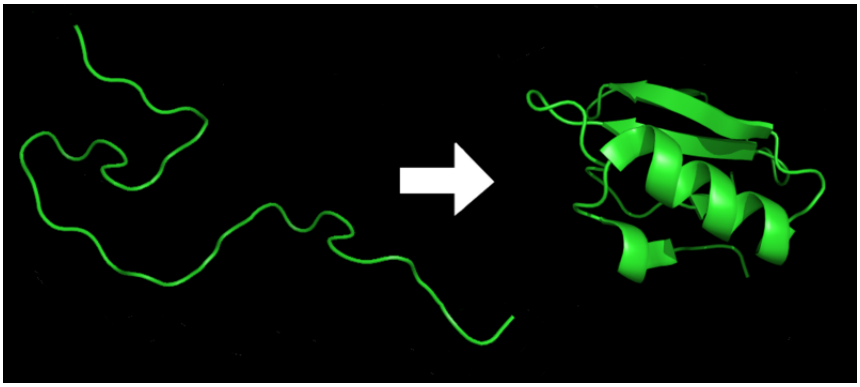


Figure 1.4: Protein Folding

from a sequence of mRNA as a linear chain of amino acids. This polypeptide lacks any developed three-dimensional structure (the left hand side of figure 1.4). However each amino acid in the chain can be thought of having certain 'gross' chemical features. These may be hydrophobic, hydrophilic, or electrically charged, for example. These interact with each other and their surroundings in the cell to produce a well-defined, three dimensional shape, the folded protein (the right hand side of figure 1.4), known as the native state. The resulting three-dimensional structure is determined by the sequence of the amino acids. The mechanism of protein folding is not completely understood.

Experimentally determining the three dimensional structure of a protein is often very difficult and expensive. However, the sequence of that protein is often known. Therefore scientists have tried to use different biophysical techniques to manually fold a protein. That is, to predict the structure of the complete protein from the sequence of the protein.

For many proteins the correct three dimensional structure is essential to function. Failure to fold into the intended shape usually produces inactive proteins with dif-

ferent properties (details found under prion). Several neurodegenerative and other diseases are believed to result from the accumulation of misfolded (incorrectly folded) proteins.

1.2 Protein Structure Prediction

Protein structure prediction is one of the most important goals pursued by bioinformatics and theoretical chemistry. Its aim is the prediction of the three-dimensional structure of proteins from their amino acid sequences, sometimes including additional relevant information such as the structures of related proteins. In other words, it deals with the prediction of a protein's tertiary structure from its primary structure. Protein structure prediction is of high importance in medicine (for example, in drug design) and biotechnology (for example, in the design of novel enzymes). Every two years, the performance of current methods is assessed in the CASP experiment.

The practical role of protein structure prediction is now more important than ever. Massive amounts of protein sequence data are produced by modern large-scale DNA sequencing efforts such as the Human Genome Project. Despite community-wide efforts in structural genomics, the output of experimentally determined protein structures – typically by time-consuming and relatively expensive X-ray crystallography or NMR spectroscopy – is lagging far behind the output of protein sequences.

A number of factors exist that make protein structure prediction a very difficult task. The two main problems are that the number of possible protein structures is extremely large, and that the physical basis of protein structural stability is not fully understood. As a result, any protein structure prediction method needs a way to explore the space of possible structures efficiently (a search strategy), and a way to identify the most plausible structure (an energy function).

In comparative structure prediction, the search space is pruned by the assumption that the protein in question adopts a structure that is reasonably close to the structure of at least one known protein. In de novo or ab initio structure prediction, no such assumption is made, which results in a much harder search problem. In

both cases, an energy function is needed to recognize the native structure, and to guide the search for the native structure. Unfortunately, the construction of such an energy function is to a great extent an open problem.

Direct simulation of protein folding in atomic detail, via methods such as molecular dynamics with a suitable energy function, is typically not tractable due to the high computational cost, despite the efforts of distributed computing projects. Therefore, most de novo structure prediction methods rely on simplified representations of the atomic structure of proteins.

The above mentioned issues apply to all proteins, including well-behaving, small, monomeric proteins. In addition, for specific proteins (such as for example multimeric proteins and disordered proteins), the following issues also arise:

- Some proteins require stabilization by additional domains or binding partners to adopt their native structure. This requirement is typically unknown in advance and difficult to handle by a prediction method.

- The tertiary structure of a native protein may not be readily formed without the aid of additional agents. For example, proteins known as chaperones are required for some proteins to properly fold. Other proteins cannot fold properly without modifications such as glycosylation.

- A particular protein may be able to assume multiple conformations depending on its chemical environment.

- The biologically active conformation may not be the most thermodynamically favorable.

Due to the increase in computer power, and especially new algorithms, much progress is being made to overcome these problems. However, routine de novo prediction of protein structures, even for small proteins, is still not achieved.

1.2.1 Ab initio protein modelling

Ab initio- or *de novo-* protein modelling methods seek to build three-dimensional protein models "from scratch", i.e., based on physical principles rather than (directly) on previously solved structures. There are many possible procedures that either attempt to mimic protein folding or apply some stochastic method to search possible solutions (i.e., global optimization of a suitable energy function). These procedures tend to require vast computational resources, and have thus only been carried out for tiny proteins. To predict protein structure de novo for larger proteins will require better algorithms and larger computational resources like those afforded by either powerful supercomputers (such as Blue Gene or MDGRAPE-3) or distributed computing (such as Folding@home, the Human Proteome Folding Project and Rosetta@Home). Although these computational barriers are vast, the potential benefits of structural genomics (by predicted or experimental methods) make ab initio structure prediction an active research field.

1.2.2 Comparative protein modelling

Comparative protein modelling uses previously solved structures as starting points, or templates. This is effective because it appears that although the number of actual proteins is vast, there is a limited set of tertiary structural motifs to which most proteins belong. It has been suggested that there are only around 2000 distinct protein folds in nature, though there are many millions of different proteins.

These methods may also be split into two groups:

- Homology modelling is based on the reasonable assumption that two homologous proteins will share very similar structures. Because a protein's fold is more evolutionarily conserved than its amino acid sequence, a target sequence can be modelled with reasonable accuracy on a very distantly related template, provided that the relationship between target and template can be discerned through sequence alignment. It has been suggested that the primary bottleneck in comparative modelling arises from difficulties in alignment rather than

from errors in structure prediction given a known-good alignment. Unsurprisingly, homology modelling is most accurate when the target and template have similar sequences.

- Protein threading scans the amino acid sequence of an unknown structure against a database of solved structures. In each case, a scoring function is used to assess the compatibility of the sequence to the structure, thus yielding possible three-dimensional models. This type of method is also known as 3D-1D fold recognition due to its compatibility analysis between three-dimensional structures and linear protein sequences. This method has also given rise to methods performing an inverse folding search by evaluating the compatibility of a given structure with a large database of sequences, thus predicting which sequences have the potential to produce a given fold.

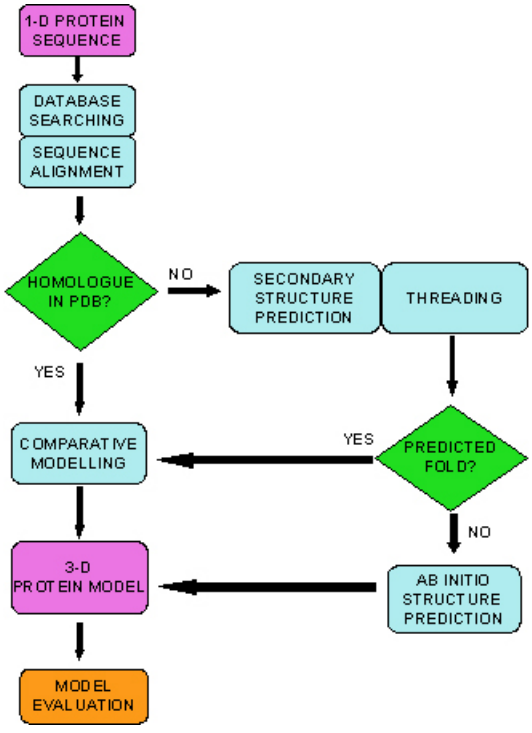


Figure 1.5: Protein Prediction

1.3 Protein Data Bank

The Protein Data Bank (PDB) is a repository for 3-D structural data of proteins and nucleic acids. These data, typically obtained by X-ray crystallography or NMR spectroscopy and submitted by biologists and biochemists from around the world, are released into the public domain, and can be accessed for free.

Founded in 1971 by Drs. Edgar Meyer and Walter Hamilton Brookhaven National Laboratory, management of the Protein Data Bank was transferred in 1998 to members of the Research Collaboratory for Structural Bioinformatics (RCSB). Rutgers University is the lead site and is currently under the direction of Helen M. Berman.

The Worldwide Protein Data Bank (wwPDB) consists of organizations that act as deposition, data processing and distribution centers for PDB data. The founding members are RCSB PDB (USA), MSD-EBI (Europe) and PDBj (Japan). The BMRB (USA) group joined the wwPDB in 2006. The mission of the wwPDB is to maintain a single Protein Data Bank Archive of macromolecular structural data that is freely and publicly available to the global community.

The PDB is a key resource in structural biology and is critical to more recent work in structural genomics.

Countless derived databases and projects have been developed to integrate and classify the PDB in terms of protein structure, protein function and protein evolution.

As of 26 September 2006, the database contained 39,051 released atomic coordinate entries (or "structures"), 35,767 of that proteins, the rest being nucleic acids, nucleic acid-protein complexes, and a few other molecules. About 5,000 new structures are released each year. Data are stored in the mmCIF format specifically developed for the purpose.

Note that the database stores information about the exact location of all atoms in a large biomolecule (although, usually without the hydrogen atoms, as their positions are more of a statistical estimate); if one is only interested in sequence data, i.e. the

list of amino acids making up a particular protein or the list of nucleotides making up a particular nucleic acid, the much larger databases from Swiss–Prot and the International Nucleotide Sequence Database Collaboration should be used.

1.3.1 File format

Through the years the PDB file format has undergone many, many changes and revisions. Its original format was dictated by the width of computer punch cards.

- PDB Format Guide – Prepared by the PDB Staff at BNL The PDB format specification can be found here, and it is vital that you read this before looking at the raw data.
- Recently PDB provides a representation of PDB data in XML format, PDBML format.
- ftp.rcsb.org The raw data can be downloaded from here.
- PDB format files can be downloaded using HTTP with URLs like this:

  http://www.pdb.org/pdb/files/4hhb.pdb.gz
- PDBML (XML) files can be downloaded using HTTP with URLs like this:

  http://www.pdb.org/pdb/files/4hhb.xml.gz
- ftp.ebi.ac.uk/pub/databases/rcsb/ Alternate download location for the PDB archive.
- www.pdb.org Statistics about the PDB can be found here.

This legacy format has caused many problems with the format, and consequently there are 'clean-up' projects;

- The Molecular Modeling DataBase (MMDB) from NCBI
- wwPDB

The MMDB uses ASN.1 (and an XML conversion of this format). The wwPDB members RCSB PDB, MSD-EBI, and PDBj are working together to make the data uniform across the archive.

Each structure published in PDB receives a four-character alphanumeric identifier, its PDB ID. This should not be used as an identifier for biomolecules, since often several structures for the same molecule (in different environments or conformations) are contained in PDB with different PDB IDs.

If a biologist submits structure data for a protein or nucleic acid, wwPDB staff reviews and annotates the entry. The data are then automatically checked for plausibility. The source code for this validation software has been released for free. The main data base accepts only experimentally derived structures, and not theoretically predicted ones (see protein structure prediction).

Various funding agencies and scientific journals now require scientists to submit their structure data to PDB.

1.3.2 Viewing the data

The structural data can be used to visualize the biomolecules with appropriate software, such as VMD, RasMol, PyMOL, Jmol, MDL Chime, QuteMol, web browser VRML plugin or any web-based software designed to visualize and analyse the protein structures such as STING. A recent desktop software addition is Sirius. The RCSB PDB website also contains resources for education, structural genomics, and related software.

Chapter 2

Computational Complexity of Protein Reconstruction from Contact Maps

2.1 Introduction

In this chapter we review the reduction proposed in [9] showing that unit disk graph recognition is NP–hard. Topic by section:

- **Graph Realization in k-Dimensional Space** (2.2)
  - Short overview of the general problem.
- **k-Sphericity with Backbone Constraint** (2.3)
  - Contact map.
  - k-Sphericity.
  - Backbone constraint.
- **A Graph that simulates Satisfiability** (2.3)
  - Reduction of SATISFIABILITY to Grid Drawing Orientability.
  - Grid Drawing Orientability Components.
- **Introducing the Backbone** (2.4)

  - From Grid Drawing Components to disjoint sequences.
  - Joining the Sequences.

- **From Orientability to 2-Sphericity with Backbone Constraint** (2.3)
  - Description of Cages
  - Description of Graph representation of Components.

2.2 Graph Realization in k-Dimensional Space

Let E_r a set of r distinct integers. A path "set" is an unordered subset P of E. A path set is "realize" in a undirected, edge-labelled tree T consisting of r edges, if each edge of T is labelled by a distinct integer from E_r, and there is a contiguous path in T whose labels consist of the integers in P. Note that since P is unordered, its presentation does not specify or constrain the order that those edges appear in T. In quite different terms, from the 1930's to the 1960's Whitney and Tutte and others studied and solved the following problems:

Graph Realization is the problem of constructing a tree from a set of its edge–labelled paths. More formally, Given subsets $P_1, \ldots, P_n$ of $\{0, \ldots, m-1\}$, find a tree $T = (V, E)$ with $E = \{0, \ldots, m-1\}$ such that every P_i is a path in T, or determine that no such tree exists.

2.3 k-Sphericity with Backbone Constraint

Formally, the contact map of threshold $t > 0$ of a protein 3D structure is the binary symmetric matrix M such that the entry i, j of M is 1 if and only if the residues i, j are distant less than t Å. In some sense, a contact map is the projection of the protein 3D structure on the two-dimensional plane. Conversely, a contact map of threshold t is *realized* by some set of k-dimensional Euclidean coordinates whether there is a mapping from the set of residues to K-dimensional points such that the Euclidean distance between points i, j is less than t Å if and only if the corresponding

i, j entry of the contact map is 1. The reconstruction problem from protein contact maps involves the problem to find three-dimensional realizations of protein contact maps.

In this general formulation the reconstruction problem is no less difficult than deciding the *sphericity* of a graph, which is well known to be a NP-hard problem. More in detail, any undirected graph G can be represented by a binary symmetric matrix M_G such that the i, j entry of M_G is 1 if and only if there is an edge between vertices i and j. A graph G has *sphericity* k whether there is a k-dimensional realization of M_G. The problem to determine the 2–SPHERICITY of a graph is also known as the *unit disk graph* recognition problem and it has been shown to be NP-hard by Breu and Kirkpatric in [9] by a polynomial time reduction from SATISFIABILITY [19]. Their proof can be easily modified to prove that also 3–SPHERICITY is NP–hard and, moreover, the authors conjecture that k–SPHERICITY is NP–hard for all $k > 1$ (graphs of sphericity 1 are known as interval graphs and they can be recognized in polynomial time [12]). Finding a realization of a graph is a different problem than determining its sphericity but this last problem can be polynomially reduced without great effort to the first one: if a polynomial-time realization algorithm would exist then it would be possible in polynomial-time to generate a set of coordinates for some graph and next check whether the coordinate set is actually a realization of the graph. Actually, in [6] the authors show that the realization of unit disk graphs cannot even approximated in polynomial-time. This result holds also for dimension three other than dimension two.

Although both three–dimensional realization and approximation are in general NP–hard problems, in this thesis we deal with the realization problem in a much more simplified setting. The problem to realize a protein 3D structure (from contact map) has much more strong constraints with respect to the general realization problem:

1. in a real 3D protein structure the residues chain forms a backbone: the residues are sequentially aligned in the backbone and the distance between a residue and its successor is always around 3.7 Å (this value is fixed for all proteins);

2. the backbone of the protein never intersect with itself.

We remark that, from a biological point of view, a realization of a contact map which does not satisfies constraints 1 and 2 is essentially useless since such realization surely does not capture the functionality of the protein.

In this chapter we actually prove that 2–SPHERICITY with constraint 1 and 2 still remains NP–hard from which follows that the realization problem with constraints 1 and 2 is NP–hard.

2.4 A Graph that simulates Satisfiability

A *k-dimensional realization* of a disk graph $G = (V, E)$ is a function $f : V \to \Re^K$ such that $(v_i, v_j) \in E$ if and only if $d(f(v_i), f(v_j)) \leq t$, where d is the Euclidean distance between two points, and t is the unit of distance. A graph G has *sphericity* k if it has a k-dimensional realization.

A graph with sphericity 2 is also called *unit disk graph*. In [9] the following theorem is proved.

Theorem 2.1 ([9]) *Unit disk graph recognition is NP-hard.*

The proof is a reduction from SAT [19] to 2–SPHERICITY. Our proof is inspired to this reduction. In the following section we summarize the details needed to understand our result.

The reduction is done in three steps. First a graph G_C^{SAT} corresponding to some C instance of SAT is constructed. Such graph is *orientable* if and only if C is satisfiable. Second a canonical drawing for G_C^{SAT} is defined, preserving the orientability property. Third a graph G_C is constructed expanding the G_C^{SAT} drawing, so that G_C has a 2D realization if and only if G_C^{SAT} is orientable.

We define an instance C of SAT as a pair (U, C) where $U = \{x_1, x_2, \ldots, x_m\}$ is a set of Boolean variables and $C = \{c_1, c_2, \ldots, c_n\}$ is a set of clauses over U. Each clause is a set of literals, each literal being a negated or non negated variable.

Without loss of generality we assume that each clause contains at most 3 literals and that each variable appears in at most 3 clauses [19].

We define G_C^{SAT} as a graph having a vertex for each clause, variable, and negated variable, and an edge between each literal vertex and clause vertex if the literal appears in the clause. So G_C^{SAT} is a bipartite graph, having edges only between clause vertices and literal vertices. The graph is *orientable* if edges can be directed such that:

- each clause vertex has outdegree at least 1;
- for each pair of literal vertices, corresponding to the same variable negated and unnegated, either the indegree of the negated variable is 0 or the indegree of the unnegated variable is 0.

In [9] has been proved that C is satisfiable if and only if G_C^{SAT} is orientable. As an example see Table 2.1 Figure (a), in which clauses and literals are linked to each other by path for the instance $C = (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$. We see in the drawing that for this assignment each clause has out-degree at most 1 and for each pair of literals at least one has in-degree 0.

In Table 2.1 (a) we show also the canonical drawing on a grid. The size of the grid is $(6|U| + 1) \times (3|C| + 2)$, each vertex associated to a square which is either empty or contain a single *component* of the drawing. There are three type of components: communication components, literals and clauses. Communications components are used to draw the edges of the graph, literals and clauses for the vertices. Each communication component can be oriented to draw directed edges. Literals and clauses components may be oriented to draw edges going in or out from the corresponding vertex. This drawing corresponds exactly to G_C^{SAT}, so that a grid drawing is orientable if and only if the underlying graph is orientable [9].

2.5 Introducing the backbone

The backbone is introduced in the following way:

- The crossover component is realized as two different sequences(one left to top, the other bottom to right) (2.1 *b*).
- The clause component is realized as three different sequences. The Literal component is realized as six different sequences (2.1 *b*).
- The literal component is realized as six different sequences (2.1 *b*).
- The other components (wire, corner) are realized as a single sequence(2.1 *b*).

Then the overall sequence is created starting by selecting at every step the bottom-rightmost sequence as shown in (2.1 *c*).

The last step is the expansion of each components to a set of vertices and edges so that the resulting graph has SPHERICITY 2 if and only if the drawing is orientable.

2.6 From Orientability to 2-Sphericity with Backbone Constraint

2.6.1 Properties of Cages

The main building block of components are cycles (called *cages*) which can contain an independent set of one, two, or three vertices (called *beads*). Adjacent cages share one edge, beads are associated to shared edges. In any realization a bead is forced to stay inside one of the two adjacent cages, eventually linking bead vertex together. As an example, in Figure 2.1 we see two cages sharing an edge with the associated independent set of two vertices (beads of size 2) staying in the left cages. For each bead two link vertices are used to keep the bead together. This two cages are oriented from right to left, the opposite orientation is obtained by reflecting beads over across the shared edge.

The maximum size of an independent set that can be embedded inside a cage is the cage *capacity*. The capacity depends only on the number of cage vertices, since the shape of the cage is decided by the realization of the graph. Cages in Figure 2.1

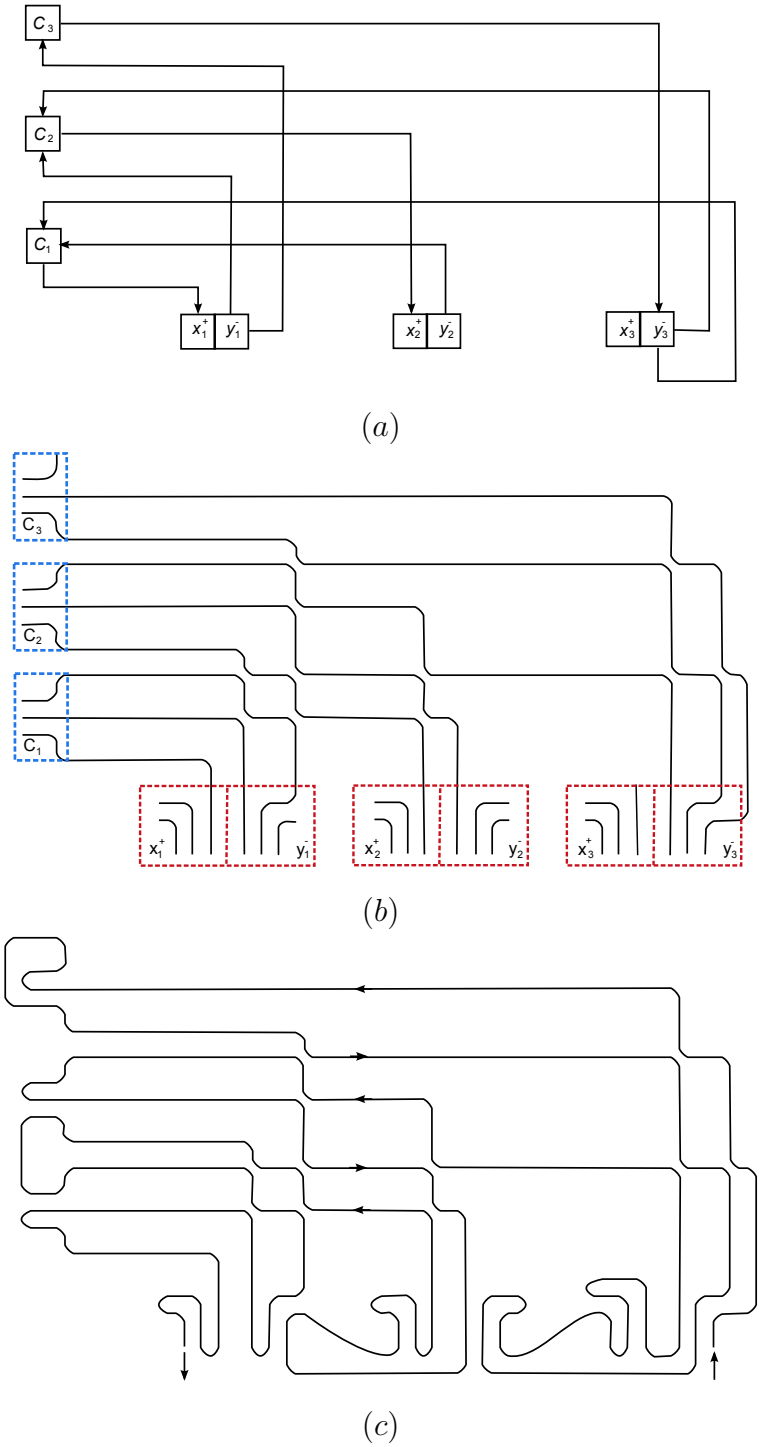


Table 2.1: (a) The graph corresponding to the SATISFIABILITY instance of $U = \{x_1, x_2, x_3\}, C = \{\{x_1, \overline{x_2}, \overline{x_3}\}, \{\overline{x_2}, \overline{x_3}\}, \{\overline{x_1}, x_2, \overline{x_3}\}\}$. (b)Introduction of the backbone constraint. The crossover component is realized as two different sequences(one left to top, the other bottom to right). The clause component is realized as three different sequences. The Literal component is realized as six different sequences. (c) The sequence is created starting by selecting at every step the bottom-rightmost sequence.

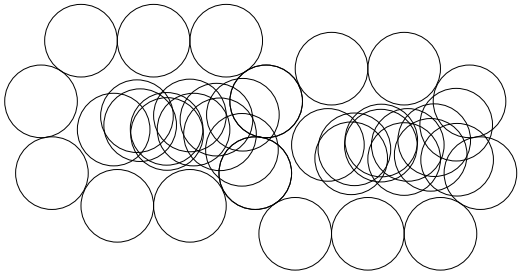


Figure 2.1: An example of adjacent oriented cages

have capacity 2, so that they may contain at most two beads of size 1 or one bead of size 2. This ensures that the two cages of Figure 2.1 may be oriented only in 3 ways: from left to right, from right to left and with beads outside cages both on the right and on the left. This allows edges to be oriented in the two classical direction and to enter both its adjacent nodes. These added degree of liberty does not influence the proof because the definition of orientability requires edges only to exit from nodes, never to enter them. For our reduction cages of capacity 1 or 2 are required. The number of vertices needed for such cages can be verified by optimal disk packing (Figure 2.2). As an example an optimal disk packing containing 2 (adjacent) disks may contain at most one disk non adjacent to the others (a bead of size 1).

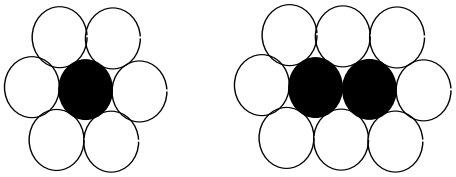


Figure 2.2: Optimal Disk Packing

For the sake of brevity we give the rest of the proof directly for contact maps. Further details on the proof regarding unit disk graphs are in [9]. In the appendix 5.8 we show the drawings relative to each 2-cages used in our realization.

A k-dimensional realization $f : V \to \mathbb{R}^k$ of a graph $G = (V, E)$ admits a *backbone* if:

1. There is an equidistant sequentialization of the vertices. That is, there is a permutation function $g : \{1, .., |V|\} \rightarrow \{1, ..., |V|\}$ of the vertices $v_i \in V$ such that $d(f(v_{g(i)}), f(v_{g(i+1)})) = n$, for every $i \in [0, .., |V \vdash 1]$ and for some constant $0 < n < 1$;

2. Such sequentialization of the vertices does not intersect with itself. That is, for every $0 \leq i, j < |V|$, the segment described by points $(f(v_{g(i)}), f(v_{g(i+1)}))$ must not intersect with the segment $(f(v_{g(j)}), f(v_{g(j+1)}))$ in the plane.

k-SPHERICITY with backbone
Instance: Graph $G = (V, E)$, positive integer k
Question: Does G have a k–dimensional realization which admits a backbone?

Theorem 2.2 *2-SPHERICITY with backbone is NP-hard.*

In the following subsections we provide a proof for Theorem 2.2 by showing the realization of each drawing grid component. We show that a graph admits a 2-dimensional realization with a backbone if and only if it is a unit disk graph. In particular, in the following subsections we will describe in detail our components and their sequentializations. Except for the literal and crossover components, all other components are essentially those described in [9] with the only difference that ours contain more points in order to satisfy the equidistant sequentialization constraint. Although our crossover and literal components are largely different from the one designed in [9], the logical structure remains inaltered with respect to the original one.

2.6.2 Wire Component

A Wire Component (see Figure 2.2) connects the various components of the grid drawing to one another and hence serves as the *directed path* between them. A wire consists of a sequence of six 1-cages hooked together, with a bead at each connecting link. It can be horizontal or vertical. It is made by a single sequence.

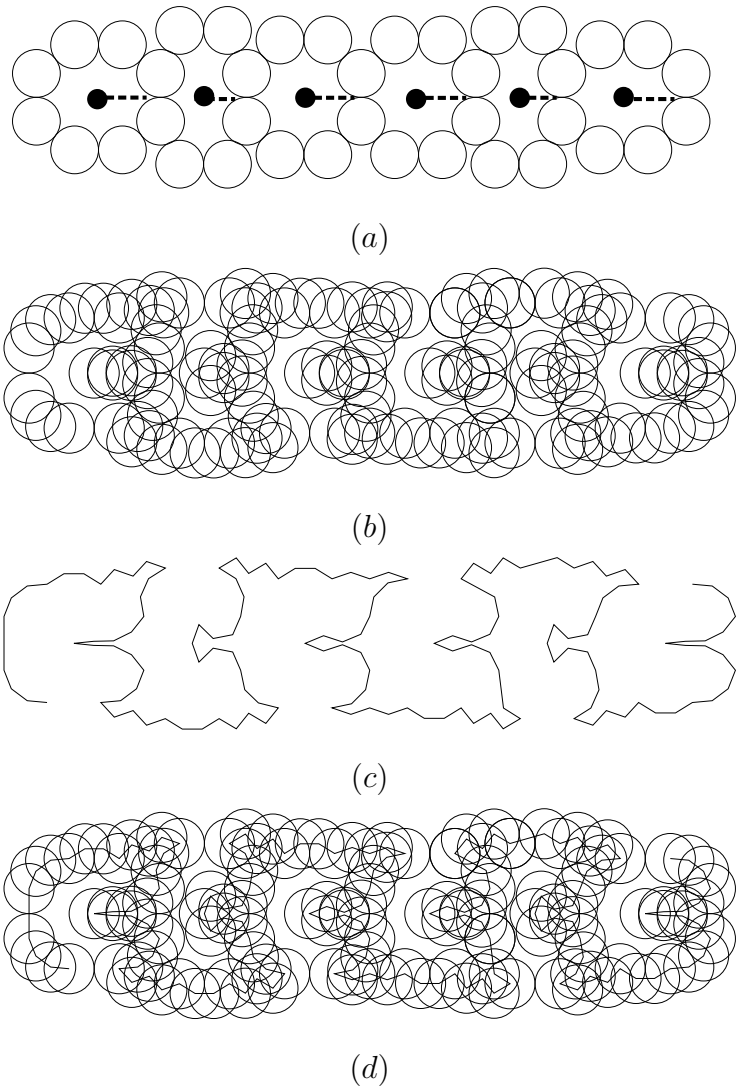


Table 2.2: Wire Component: (a) Details on Bead; (b) Details on Contacts; (c) Details on Sequence; (d) Details on both contacts and sequence.

2.6.3 Corner Component

A Corner Component (see Figure 2.3) connects the various components of the grid drawing to one another and hence serves as the *directed path* between them. A corner consists of a sequence of six 1-cages hooked together, with a bead at each connecting link. It can be rotated with an angle of $\pi/2$ to have each one of its four configurations. It is made by a single sequence.

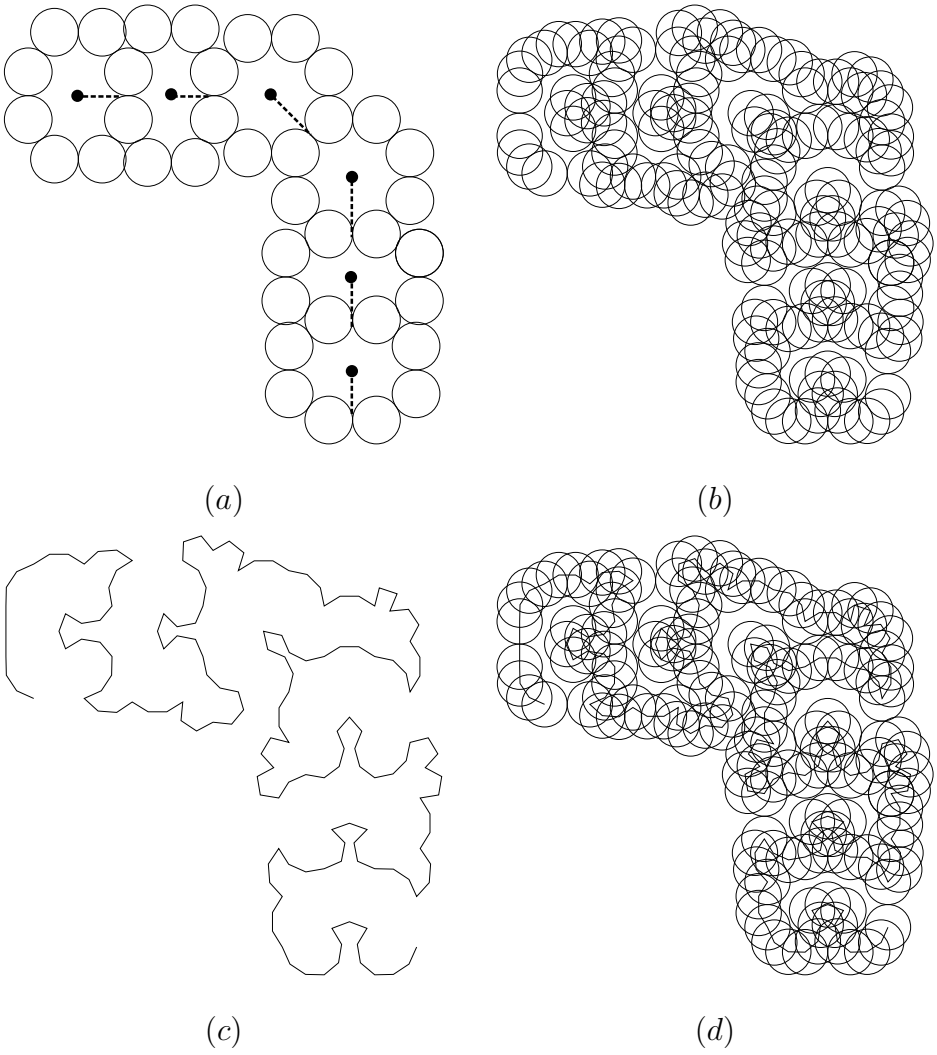
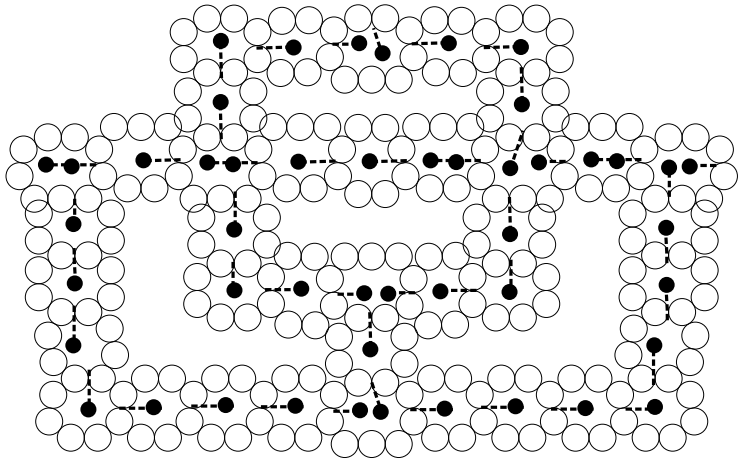
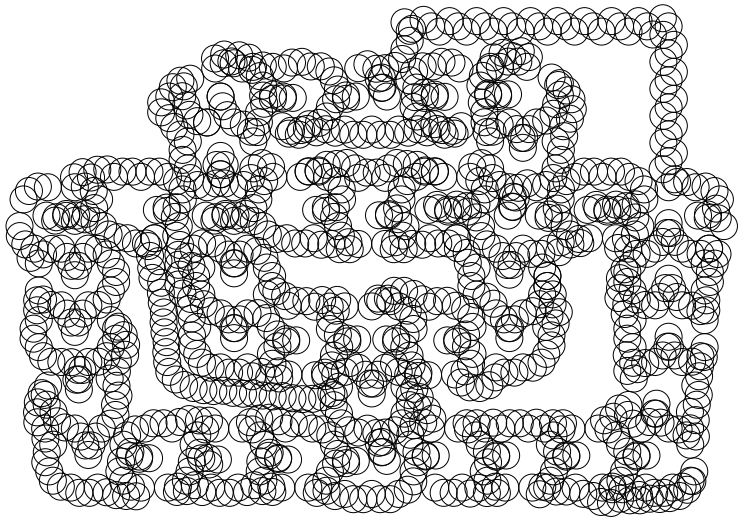


Table 2.3: Wire Component: (a) Details on Bead; (b) Details on Contacts; (c) Details on Sequence; (d) Details on both contacts and sequence.

2.6.4 Crossover Component



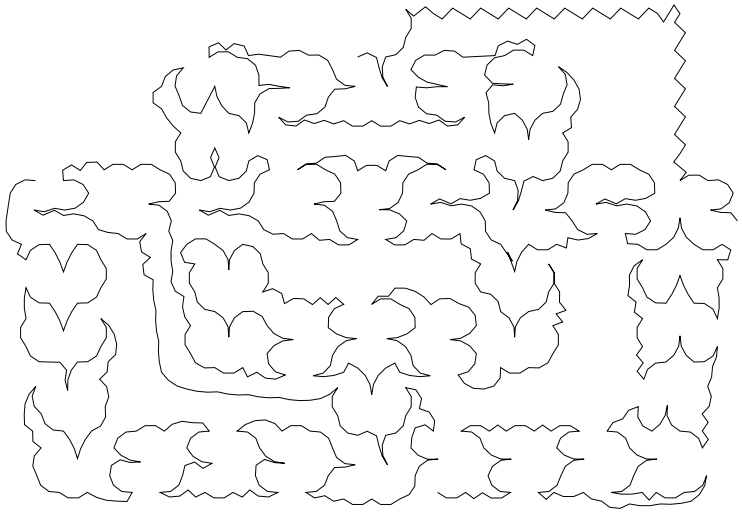
(a)



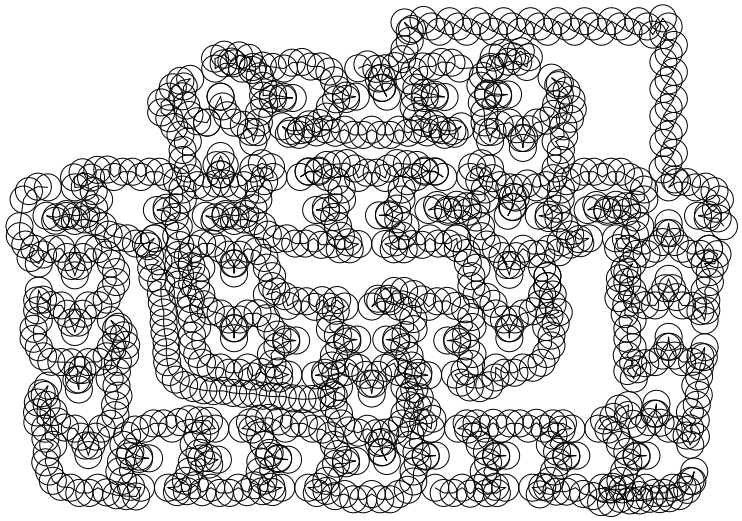
(b)

Table 2.4: Crossover Component: (a) Details on Bead; (b) Details on Contacts.

The most challenging component is the crossing component (see Table 2.4 and Table 2.5 - the orientation of the crossover chosen in the four figure is from right to left and from top to bottom). The crossing components ensures that if the T_s (respectively T_n,T_e and T_w component) is oriented towards the component than T_n (respectively T_s,T_w and T_e component)is oriented away from it. However it is



(c)



(d)

Table 2.5: CROSSOVER component: (c) Details on Sequence; (d) Details on both contacts and sequence.

possible that both component are oriented away from the component (see Figure 2.3). The crossover component is made by several 1-cages and 2-cages to guarantee a correct behavior. It is composed by two sequences, one going from the right to the top, and one from the bottom to the left. It is important not to confuse the such sequences, which are ever the same, with the orientation which depends from the Grid Drawing.

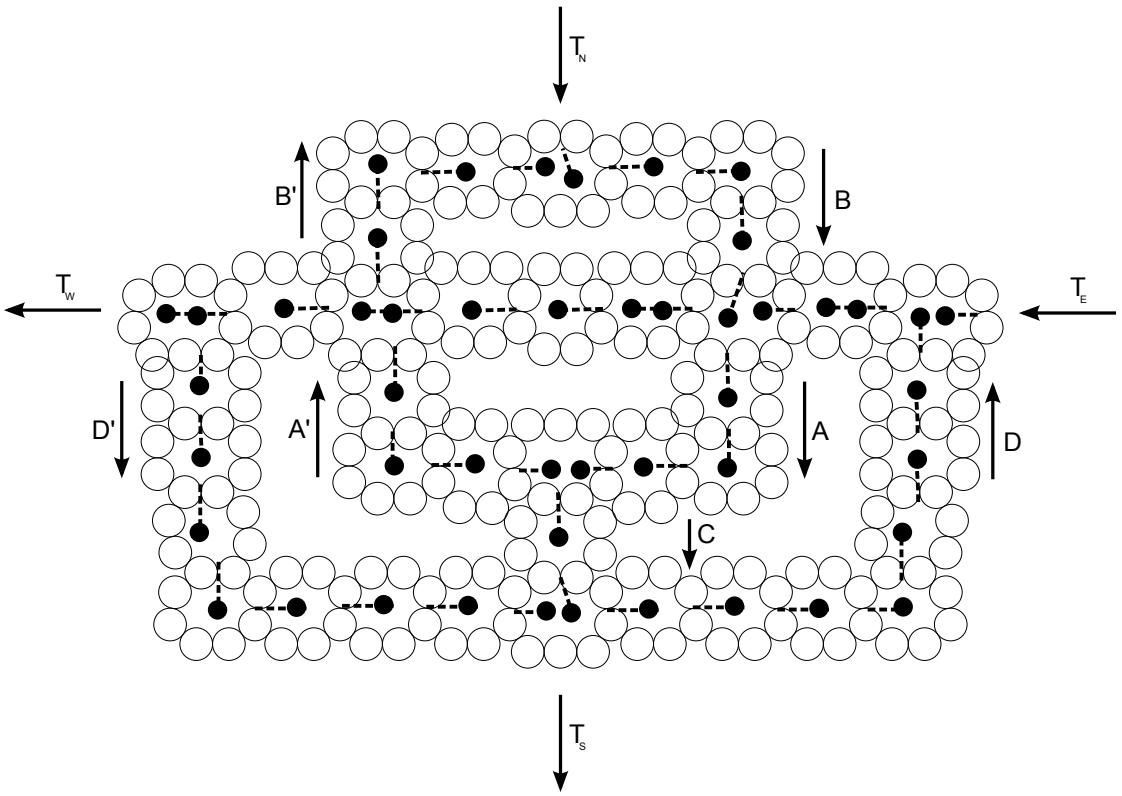


Figure 2.3: Cross Consistency

Let H denote the horizontal path between T_w and T_e. Let $d(r) :\uparrow$ $d(r) :\downarrow$ denote that wire r is directed in northern and southern direction, respectively. In the example, the following properties holds: $d(A), d(A'), d(B), d(C), d(D') :\downarrow$ and $d(B'), d(D') :\uparrow$. Let's start with the simpler, horizontal direction. Assume that T_e is directed towards the component, as shown in figure. All chains in H must point to

the left(because each cage has capacity 2) and it follows that T_w must be directed away from the crossing. By symmetry, the same holds for the opposite direction. Now, let's consider the vertical direction and assume T_s is directed towards the component.It follows that or $d(D) :\uparrow$ or $d(D') :\downarrow$. It's not possible that both D and D' are directed north because that would result on a contradiction in H. Whether D or D' is directed towards or away from the component is imposed by the direction of the horizontal terminals. In case both horizontal terminals are directed away from the component, a realization arbitrarily choose one of the two. Assuming without loss of generality that $d(D) :\downarrow$ and $d(D') :\uparrow$, it follows that $d(B') :\downarrow$ and $d(A') :\uparrow$. Combining $d(B') :\downarrow$ with $d(C) :\uparrow$ yields $d(B) :\uparrow$ and consequently $d(A) :\uparrow$. By $d(A) :\uparrow$ and $d(A') :\uparrow$, we have that T_n is directed away from the crossing.

Last, it's must be shown that the crossing works if T_n is directed towards the component. Again it is not possible to have both $d(D) :\uparrow$ and $d(D') :\uparrow$. In case $d(D) :\uparrow$ and $d(D') :\uparrow$ it's clear that the direction of T_s is south. Hence, it remains to analyze the cases for which exactly one of D and D' is directed upwards(again a horizontal terminal decides which of the two is directed upwards). Assuming without loss of generality that $d(D) :\downarrow$ and $d(D') :\uparrow$ forces the part of H between the two outermost cages to be directed towards the west and therefore, it follows $d(B') :\downarrow$ and $d(A') :\uparrow$. $d(A') :\uparrow$ and $d(T_n) :\downarrow$ leads to $d(A) :\downarrow$ and consequently $d(B) :\downarrow$. In combination with $d(B') :\downarrow$ this results in $d(C) :\downarrow$, and finally $d(T_s) :\downarrow$. The case $d(D) :\uparrow$ and $d(D') :\downarrow$ is symmetric.

2.6.5 Truth Setter Component

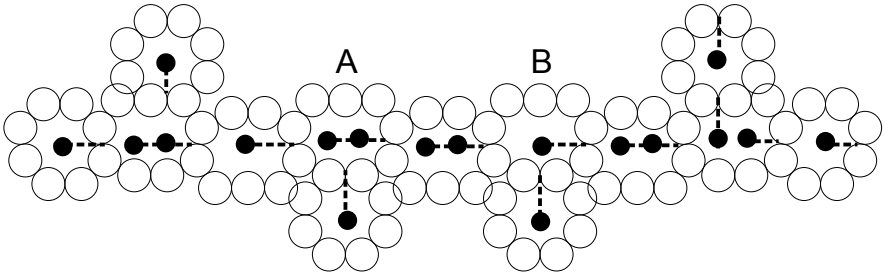
The truth setter component (see Table 2.6) contains the two literal components. It must guarantee that all terminals are directed away from one of its literal components, i.e.: that the variable it represents can be set either true or false. Assume that one negative terminal (i.e.: on the right side) is directed towards the component, it follows that at least one single-chain is embedded in cage B thus forcing the double chain between A and B into A. This consequently forces all positive terminals to

be directed away from the terminal. By symmetry, the same holds in the other direction as well.

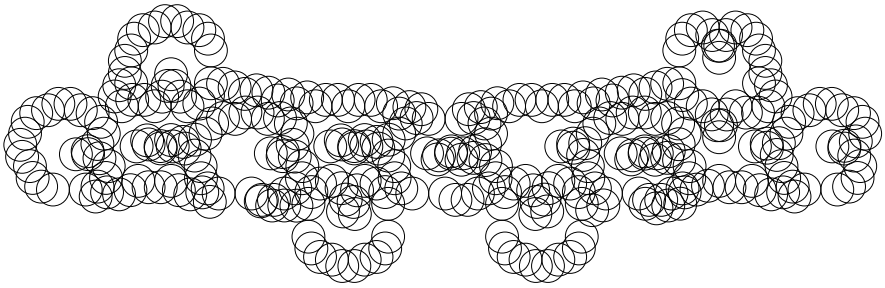
2.6.6 Clause Component

A Clause Component (see Figure 2.7) connects the various components of the grid drawing to one another and hence serves as the *directed path* between them. A clause consists of a sequence of nine 1-cages hooked together, with a bead at each connecting link. It can be horizontal or vertical. It is made by a single sequence.

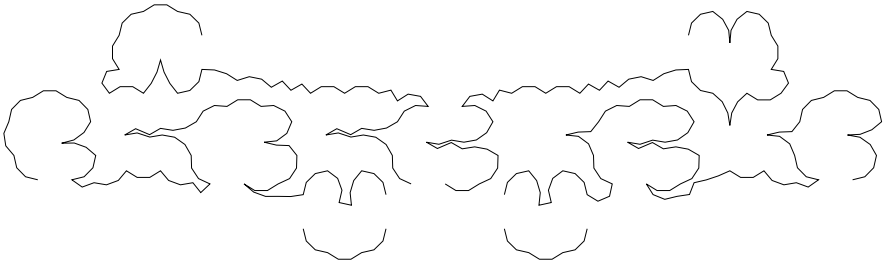
The central cage A, can contain exactly two of the adjacent bead. Hence, in accordance to the requirements imposed by G_C^{SAT}, at least one terminal must be directed away from the component. In case a clause contains less than three variables, some of the terminals may be *capped* (see[9]).



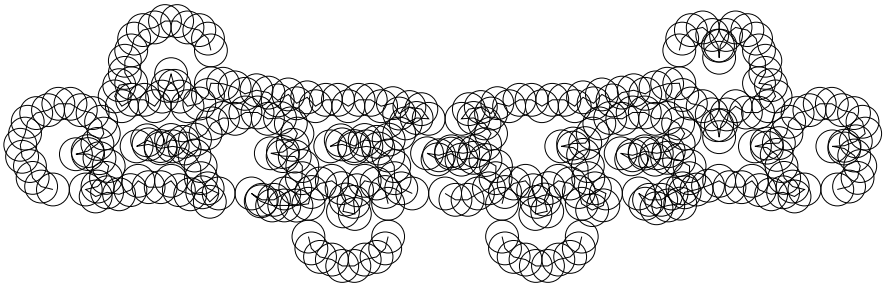
(a)



(b)



(c)



(d)

Table 2.6: Truth Setter Component. The left hand represent the positive literal, the right half represent the negative literal. All terminals of the positive literals are directed away of the component.

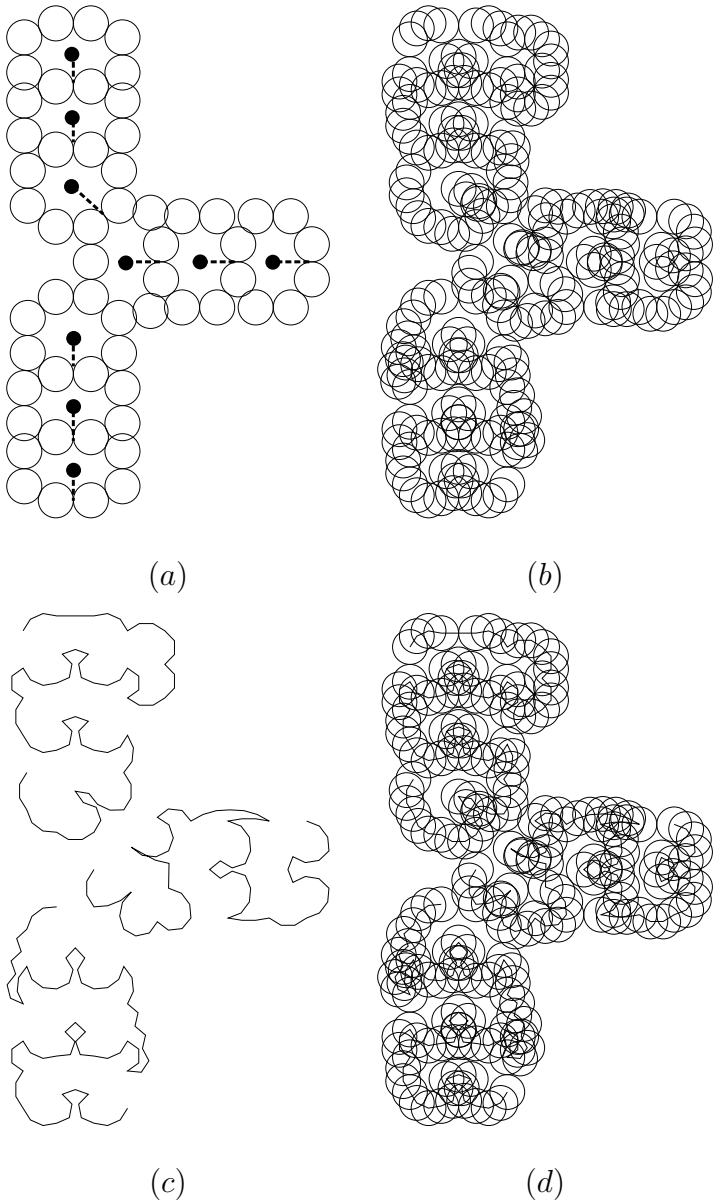


Table 2.7: Wire Component: (a) Details on Bead; (b) Details on Contacts; (c) Details on Sequence; (d) Details on both contacts and sequence.

Part II

Experimental Results

Chapter 3

Protein Structure Reconstruction from Contact Maps

Bioinformatics is an emerging field undergoing rapid, exciting growth. This has been mainly fueled by advances in DNA sequencing and mapping techniques. The Human Genome Project (see [3, 4, 14]) resulted in an exponentially growing database of genetic sequences, while the Structural Genomic Initiatives (see [2, 22, 21]) is doing the same for the protein structure database. One of the grand challenges in bioinformatics is protein structure prediction, where one is interested in determining the 3D structure of a protein given its aminoacid sequence. It is well know that proteins fold spontaneously and reproducibly to a unique 3D structure in aqueous solution.

3.1 Contact Maps

Proteins structures are described by the coordinates of the atoms that concur to constitute the macromolecules. For a protein with n atoms we need $3 \cdot n$ numbers to specify its three-dimensional (3D) structure. An alternative is to consider the distance matrix. The distance matrix is a symmetric matrix that contains in its cells the Euclidean distance between each pair of atoms. If the number of atoms is n we need n^2 elements; since the matrix is symmetric (the distance between atoms i and j is the same of that between j and i) the effective number of needed elements

is only $n \cdot \frac{(n-1)}{2}$. In order to simplify the protein representation, not all protein atoms are taken into account and residues are considered as unique entities. In this case the distance matrix has a number of rows (and columns) equal to the residue numbers. Each distance matrix entry is then the distance between residue i and j. The distance between two residues can be defined in different ways, such as:

- the distance between a specific pair of atoms (i.e. $C_\alpha - C_\alpha$ or $C_\beta - C_\beta$);
- the shortest distance among the atoms belonging to residue i and those belonging to residue j;
- the distance between the centers of mass of the two residues.

Starting from the protein distance matrix and selecting an arbitrary distance cut-off (threshold), a further simplified representation can be obtained: the protein contact map. Residues are in contact if their distance is less than or equal to the pre-assigned threshold. Contact maps are binary symmetric matrices, whose elements different from 0 (and set to 1) represent the contacts between residues. In our work we have used the C_α representation of the protein backbone, and for sake of simplicity we refer to the protein C_α trace as "protein structure" or 3D protein structure.

3.2 Protein representation with Contact Maps

The 3D conformation of a protein may be compactly represented in a symmetrical, square, boolean matrix of pairwise, inter-residue contacts, or "contact map". The contact map provides a host of useful information about the protein's structure. For example clusters of contacts represent certain secondary structures and also capture non-local interactions, giving clue to the tertiary structure. We adopt the widely used C representation of the protein backbone, where residues are considered as unique entities. The contact map of a given protein is a binary symmetric matrix CM such that $CM[i, j] = 1$ if and only if the Euclidean distance between residues i and j is less than or equal to a pre-assigned threshold t (Figure 3.1).

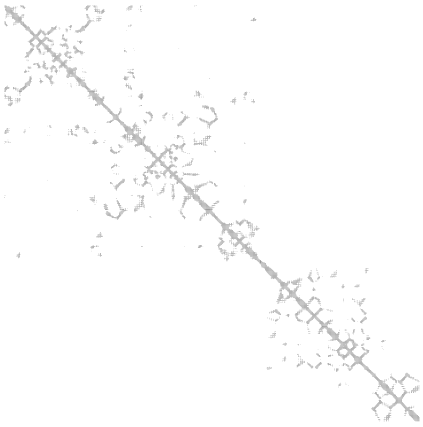


Figure 3.1: Average RMSD values on the different SCOP classes as obtained using contact maps computed with a threshold of 13 Å

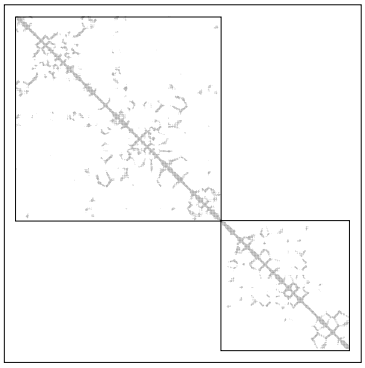


Figure 3.2: Average RMSD values on the different SCOP classes as obtained using contact maps computed with a threshold of 13 Å. Details on the double domains structure of the protein showed in the contact map.

Typical values of t considered in literature vary between 7 and 12 Å. In general , higher threshold values allow better reconstruction, and in this example we adopt t = 12 Å. To measure the similarity between two three-dimensional protein structures, described by some set of coordinates $C, C' \in \mathbb{R}^{3n}$, we use the Root Mean Square Deviation (RMSD); it is defined as the smallest distance $D_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(C^n[i] - C_k[i])^2}$, where $C_k \in \mathbb{R}^{3n}$ is obtained by rotating and translating the coordinates set C.

3.2.1 Reconstruction with Contact Maps

Distance geometry (see [1]) deals with the characterization of mathematical properties that can be derived from distance values between pairs of points. The mathematical foundation of distance geometry is essentially due to Cayley (1841) and Menger (1928) who show how some basic geometry properties, such as convexity, could be defined in terms of distance values. Distance geometry is the mathematical basis for a geometric theory of molecular conformation [8]. A distance geometry description of a molecular system consists of a list of distances and chirality constraints. These are, respectively, lower and upper bounds on the distances between pairs of atoms, and the chirality of its rigid quadruples of atoms. The distance geometry approach is based on the assumption that is possible to adequately define the set of all possible conformations, or conformation space, of just about any non rigid molecular system by means of such purely geometric constraints. Distance geometry also plays an important role in the development of computational methods for analyzing distance geometry descriptions. The goal of this calculations is to determine the global properties of the entire conformation space, as opposed to the the local properties of individual members. This is done by deriving new geometric fact about the system from those given explicitly by constraints. Although numerous constraints can be derived from knowledge of molecular formulas, in many cases additional non-covalent constraints are needed in order to define precisely the accessible conformation space. These must be obtained from additional experiments, and thus one of the best-known application of distance geometry is the determina-

tion of molecular conformation from experimental data, most notably NMR spectroscopy. One fundamental problem in distance geometry is to find a correct set of three-dimensional Euclidean coordinates that satisfy a set of distance constraints. In general, a set of points in the three-dimensional space that satisfy some given coordinates does not exist. However, Cayley and Menger gave necessary and sufficient conditions for a set of positive values to be the exact distances between pairs of three-dimensional coordinates. Thus, given a consistent set of distances in the three-dimensional space, the problem to find coordinates which satisfy such exact distance constraints can be solved by polynomial time algorithm [1], while the problem is NP-hard when the given set of coordinates is sparse [23]. NMR spectroscopy and X-ray christallography are the most widely used experimental techniques to obtain bounds to the inter atomics distances rather than exact values. The distance geometry based approach to the protein structure reconstruction problem aims at developing techniques to recover the 3D protein structure, given a set of upper and lower bounds to residue inter atomic distances. The problem of computing a set of consistent coordinates is in general intractable [11]. Havel and Crippen developed a recovering algorithmic technique from a sparse set of lower and upper bounds to the inter-atomic distances [8, 10]. Their algorithmic first uses some bound smoothing techniques to estimate bounds values to the missing distances. Then it uses an algebraic technique known as EMBED algorithm to generate an approximate set of three dimensional coordinates adopted as a starting solution for an optimization technique.

Chapter 4

COMAR
An Heuristic for Reconstruction of 3D Structure from Contact Maps

4.1 Introduction

In this chapter we present COMAR (Contact Map Reconstruction) [16]. First we introduce the heuristic and we show its main routines with the relative behavior, then we show the experimental results.

4.2 How it Works - Initial Solution

COMAR finds a set of three-dimensional coordinates consistent with some native contact maps. COMAR consists of two phases (see the pseudo code below). In the first phase it generates an initial set of 3D coordinates $C \in \mathbb{R}^{3\times n}$ while in the second phase it refines iteratively the set of coordinates by applying a correction/perturbation procedure to C. The refinement applies until the set of coordinates is consistent with the given contact map or until a control parameter becomes 0. The control parameter has initially a positive value and it is decremented every some amount of refinement steps. If the parameter reaches 0 and a correct solution

is still not found, a new initial random solution is generated and the refinement process starts over again.

COMAR($CM \in \{0,1\}^{n\ timesn}, t \in N$)

```
 1  repeat
            ▷ Phase 1: Initial Solution
 2          C ← RANDOM-PREDICT(CM,t)
            ▷ Phase 2: Refinement
 3          C ← Correct(CM,C,t)
 4          set ε to strictly positive value
 5          while C is not consistent with CM and ε > 0
 6               do C ← Perturbate(CM,C,t,ε)
 7                  C ← Correct(CM,C,t)
 8                  decrement slightly ε
 9     until C is consistent with CM
10  return C
```

4.2.1 Generation

RANDOM-PREDICT($CM \in \{0,1\}^{n \times n}, t \in N$)

```
1  {CM_1, ..., CM_k} ← SPLIT(CM)
2  for i ← 1 to k
        do
3          C_i ← EMBED (GUESS-DIST(CM_I,t))
4  C ← MERGE(C_1, ..., C_k, CM)
5  return C
```

The computing of the initial solution is preceded by a scanning of the contact map for the existence of splittable components (**SPLIT**). Splitting the initial contact map in submatrices is done to locate those fragments of proteins which demonstrates an high degree of independence with mutual interactions. The submatrices are then separately used to create a set of coordinates (**EMBED**) to be merged (**MERGE**)

in an initial solution. The merging procedure is managed by selecting, between a set of equally distributed three-dimensional angles, the best rotation of coordinates corresponding to each component with respect to the lower number of errors generated in the contact map.

4.2.2 SPLIT

The SPLIT procedure splits a native contact map in submatrices in relation to those fragments of the protein which shows a high degree of independence with respect to mutual interactions. In other words, we identify submatrices of the contact map such that their residues have no contact outside the submatrix itself. In searching these submatrices we ignore contacts near the main diagonal, since each residue is in contact with the residues close to it in the protein chain. So, we call thickness the minimum distance from the main diagonal of a contact to be considered in the splitting procedure. Formally we say that a contact map matrix $CM \in \{0,1\}^{n\times n}$ is splittable with thickness T in the two submatrices $CM_{1,j} \in \{0,1\}^{j\times j}$ and $CM_{1,j} \in \{0,1\}^{n-j+1\times n-j+1}$ if and only if $CM[h,k] = 0 \forall h \in [1,j], k \in [j,n]$ such that $|h-k| \geq T$. Given a contact map CM the **SPLIT** function determines if it is splittable and return the two submatrices. First it calculates the number of contact shared by residues before and after each position in the sequence of residues (**SPLICE-CREATION**). Then divides CM in submatrices of size at least *Accepted Size* sharing no contacts with other submatrices:

- having no contacts besides the ones near the main diagonal, allowed by thickness T and denoted by a sequence of *0s* in the array of shared residues V (line 7);

- sharing no contacts with neighbor submatrices, denoted by a sequence of values preceded and followed by a 0 in the array of shared residues V(line 8).

```
SPLIT(CM ∈ {0,1}^{n×n})
 1  V ← SPLICE-CREATION(CM)
 2  AcceptedSize ← 13
 3  s ← 1
 4  D ← {}
 5  for i ← 1 to n
        do
 6          if (i − s > AcceptedSize)
              then
 7                  if (V[k] = 0 ∀k ∈ [s − 1, i − 1] and V[s], V[i] ≠ 0) or V[s] = 0 and V[i] = 0
                      then
 8                          D ← D∪{submatrix of CM from s to i}
 9                          s ← i
10  return D
```

For each position $i \in [1, n]$ **SPLICE CREATION** counts the number of contacts in the rectangular submatrix of CM having lower left corner at position i on the main diagonal (line $i-1$, row $i+1$) and the same upper right corner of CM (line 0,row n). Contacts in the lower left corner of this rectangular submatrix, having position j, k such that $|j-k| \leq T$, are not considered (line 6). The *thickness* parameter T is initialized as the mean over all residues of the column of the first 0 found starting from the main diagonal.

```
SPLICE-CREATION(CM ∈ {0,1}^{n×n})
1  V ← {}
2  for i ← 1 to n
       do
3          V[i] ← 0
4          for k ← 1 to n
               do
5                  if |j − k| > T
                       then
6                          V[i] ← V[i] + CM[j,k]
7  return V
```

4.2.3 MERGE

The MERGE procedure tries to merge coordinates $C_1, \ldots, C_k$ (each one constructed by the corresponding submatrix splitted from the contact map CM) into a structure consistent with the whole contact map CM. The merging process is performed incrementally (lines 3–16), adding at each step i the set of coordinates C_i to the resulting structure C. The **TRANSLATE** procedure (line 4) translates the coordinates in C_i to superimpose the common residue between C_i and the already build structure. The 50 random rotations of C_i are generated. The best rotation is selected as the one for which the contact map of the current structure has the minimum number of differences with the corresponding submatrix of original contact map (line 5–15). The **RANDOM** procedure generates three random numbers in the intervals specified. The **ROTATE** $(C_i, \{x, y, z\})$ function returns the rotation of the set of coordinate C_i over the three principal axes by angles $\{x, y, z\}$.

```
MERGE(C_1 ∈ R^{3×n_1}, ..., C_k ∈ R^{3×n_k}, CM ∈ {0,1}^{n×n})
    Require n_1 + ... + n_k = n
 1  C ← {}
 2  e ← ∞
 3  for i ← 1 to k
        do
 4      C_old ← C
 5      C_i ← TRANSLATE(C_i, C)
 6      for j ← 1 to 50
            do
 7          [x, y, z] ← RANDOM([0, π], [−π, π], [π, π])
 8          C_i' ← ROTATE(C_i, {x, y, z})
 9          C' ← append C_i' to C_old
10          CM' ← contact map of C'
11          e' ← differences between CM' and CM
12          if e > e'
              then
13                e ← e'
14                C ← C'
15  return C
```

A fast and reliable way to obtain good starting coordinates for the splittable components is provided by the matrix embedding algorithm, that can be used to compute a set of three-dimensional coordinates that is, in a certain sense, the best three-dimensional fit for some distance matrix D. By using some a priori knowledge about the physical conformation of the proteins, the GUESS-DIST procedure tries to guess a possible set of distances $D \in R^{3\times n}$ consistent with some native contact map $D \in \{0,1\}^{3\times n}$. Generally, no set of three-dimensional points is consistent with some distance matrix D. However, EMBED use standard numerical linear algebra methods to find the least distorted projection of D in the three-dimensional Euclidean space.

4.2.4 GUESS-DIST

The **GUESS-DIST** procedure tries to guess a possible set of distances $D \in R^{n\times n}$ consistent with some $CM \in \{0,1\}^{n\times n}$ of threshold t by using some a priori knowledge about the physical conformation of the proteins. For instance, residues that form the backbone of a protein are usually placed according to the typical distance value of 3.8 Å(the C_α-C_α distance). Other typical distance values can be obtained experimentally from the real proteins. The set of experimental typical values used by the **GUESS-DIST** procedure are collected in **DISTANCE**, which returns a random typical value for every couple of residues i, j and threshold t. The RANDOM procedure generates a random number in the interval specified.

DISTANCE($t \in N$,$i \in N$,$j \in N$)

Require $1 \leq i, j \geq n$

1 **if** $|i-j| = 0$
then
return 0

2 **if** $|i-j| = 1$
then
return 3.8

3 **if** $|i-j| = 2$
then
return 6 + RANDOM($[-1.5, 1.5]$)

4 **if** $|i-j| = 3$
then
return MAX (0,7.5 + RANDOM($[7.5-t, t-7.5]$))

5 **if** $|i-j| > 3$
then
return $(\frac{0.91-t}{100})t$+ RANDOM($[-t+(\frac{0.91-t}{100})t, t+(\frac{0.91-t}{100})t]$)

Any set of distances D must satisfy the triangle inequality, in order to be three-dimensional consistent. To obtain from D a set of guessed distances that satisfy the

triangle inequality, we run the standard **SHORTEST-PATH** algorithm, on the weighted graph identified by the D matrix.

```
GUESS-DISTANCE(CM ∈ {0,1}^{n×n}, t ∈ N)
1  for i ← 1 to n
       do
2      for j ← 1 to n
           do
3          if CM[i,j] = 1
             then
4                D[i,j] ← DISTANCE(t,i,j)
             else
5                D[i,j] ← ∞
6                D[j,i] ← D[i,j]
7  return SHORTEST-PATH(D)
```

4.2.5 Independent Area Identification

4.3 How it Works - Refinement

The second step of the algorithm applies iteratively a local correction/perturbation heuristic technique to the randomly predicted set of coordinates to obtain a new set of coordinates closer to the native contact map. We call *not well placed* those residues whose coordinates are not consistent (according to the contact map) with the coordinates of all other residues. The local correction technique CORRECT attempts to change the coordinate of every not well placed residue i in a new coordinate which does not affect the old set of well placed residues.

4.3.1 Correction Phase

CORRECT($CM \in \{0,1\}^{n\times n}$,$C \in R^{3\times n}$,$t \in N$)

1 **for** $i \leftarrow 1$ **to** n
  do
2 **if** i is not well placed
  then
3 $C[i] \leftarrow$ MOVE(CM,C,t,i)
4 **return** C

The procedure to approximate a good and safe coordinate for some residue i is described in MOVE. It changes the coordinate $C[i]$ to the coordinate of a point on the surface of the sphere of radius i centered in $C[i]$.The point is chosen in a region of the surface which is supposed to be as distant as possible from the whole set of residues j not well placed with respect to i such that $CM[i,k] = 1$. The radius of mobility r_i of the residue i is defined as:

$$r_i = min[D_0 - t, t - D_i]$$

where

- $D_0 = min\{d_{ij} > t \text{ and } CM[i,j] = 0\}$
- $D_1 = min\{d_{ij} > t \text{ and } CM[i,j] = 1\}$

then, by definition, the coordinate $C[i]$ of the residue i can be safely changed in any coordinate $c \in R^{3\times n}$ such that $|C[i] - c| \leq r_i$ without decreasing and eventually increasing the cardinality of the set of residues well placed with respect to i.

4.3.2 Move

The MOVE procedure projects some $C[i]$ coordinate on the surface of the sphere of radius (of mobility) r_i and centered in $C[i]$. The direction of the projection is described by a vectorial pseudo-force F applied to i. For every residue j not well

placed with respect to i, let consider the vectorial pseudo-force $F_j = (\frac{C[i]-C[j]}{d_{ij}})$ of magnitude 1 and direction ij. The point on the surface of the sphere (line 12) is then identified by the pseudo force F, resulting by the vectorial addition of forces $F_j{}'$. $F_j{}' = F_j$ when $CM[i,j] = 1$; $F_j{}' = -F_j$, i.e. $F_j{}'$ has opposite direction to F_j, when $CM[i,j] = 0$.

MOVE($CM \in \{0,1\}^{n \times n}$,$C \in R^{3 \times n}$,$t \in N$, $i \in [1,n]$)

1 $r_i \leftarrow$ radius of mobility at threshold t of residue i
2 $F \leftarrow \{0,0,0\}$
3 **for** $j \leftarrow 1$ **to** n
4 **do**
5 **if** j is not well placed with respect to i
then
6 **if** $CM[i,j] = 1$
then
7 $F \leftarrow F - \frac{C[i]-C[j]}{d_{ij}}$
else
8 $F \leftarrow F + \frac{C[i]-C[j]}{d_{ij}}$
9 **return** $C[i] + F(\frac{r_i}{|F|})$

4.3.3 Perturbation Phase

A run of the correction procedure may reduce the radius of mobility for not well placed residues. In order to maintain as large as possible the radius of mobility for such residues, after a correction procedure it is applied a small perturbation to the coordinates set using the **PERTURBATE** procedure.

PERTURBATE($CM \in \{0,1\}^{n \times n}$,$C \in R^{3 \times n}$,$t \in N$,$\varepsilon \in R$)

1 **for** $i \leftarrow 1$ **to** n
  do
2 **for** $j \leftarrow 1$ **to** n
    do
3 **if** $t - \varepsilon < |C[i] - C[j]| \leq t$ and $CM[i,j] = 1$
      then
4 bring closer $C[i]$ and $C[j]$ of $\frac{\varepsilon}{10}$
5 **if** $t < |C[i] - C[j]| \leq t + \varepsilon$ and $CM[i,j] = 1$
      then
6 move away $C[i]$ and $C[j]$ of $\frac{\varepsilon}{10}$
7 **return** C

For every residue i and every residue j well placed with respect to i, if their distance d_{ij} is under the given threshold ($CM[i,j] = 1$) but close to the threshold then **PERTURBATE** changes the coordinates of i and j in order to make them a bit more closer (lines 3–5). If d_{ij} is above a given threshold ($CM[i,j] = 0$) but close to the threshold then **PERTURBATE**changes the coordinate of i and j in order to make them a bit more distant (lines 6–8). A perturbation can introduce new errors to the coordinates set, but,conversely, it avoids that not well placed residues get stuck.

4.4 Experimental Results

4.4.1 Protein set

The list of proteins with their relative structural classification were selected from SCOP release 1.67. The corresponding protein structures were downloaded from the PDB and the files with coordinates obtained with X-ray experiments (with resolution < 2.5 Å) and without missed internal residues are the only ones to be retained. Then using BLAST [24] sequence redundancies are removed, ending with a dataset

of 1760 protein chains with sequence similarity lower than 25%. The distribution of the 1760 protein chains accordingly to SCOP is shown in Figure 4.1 The protein set resulting contains 1502 one-domain and 258 multi-domains chains. The complete set is available at the web site `http://vassura.web.cs.unibo.it/protlist.tgz`.

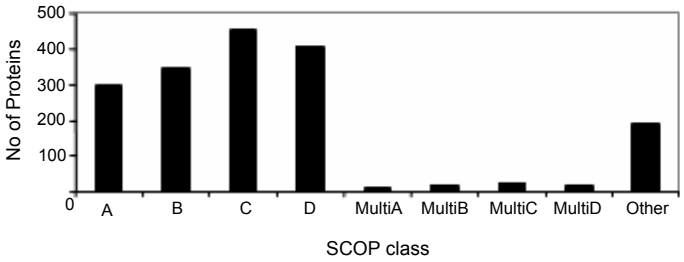


Figure 4.1: Distribution of our protein set according to the SCOP classes. A=all alpha; B=all beta; C=Alpha/Beta; D=Alpha+Beta. Multi-$\{A, B, C, D\}$ and Other contain multi-domain proteins.

4.4.2 Hardware Configuration

All the test runs are executed on personal computers equipped with an Intel Pentium 4 processor with a clock rate of 2.8 GHz and 1 Gb of RAM memory. Times reported are measured using the time() C library function. During each run the program collects time information before reading the input and again after computing the result; the CPU time actually elapsed is computed as the difference between the two figures.

4.4.3 COMAR Convergence

The termination conditions let the algorithm run until a set of coordinates consistent with a given input contact map is found. Formally, given a contact map CM and a set of coordinates C, COMAR *converge* into C when the refinement of C leads to a new set of coordinates consistent with CM. COMAR refinement is more likely to

converge as soon as the 3D structure described by the initially guessed coordinate set is sufficiently similar to the native one.

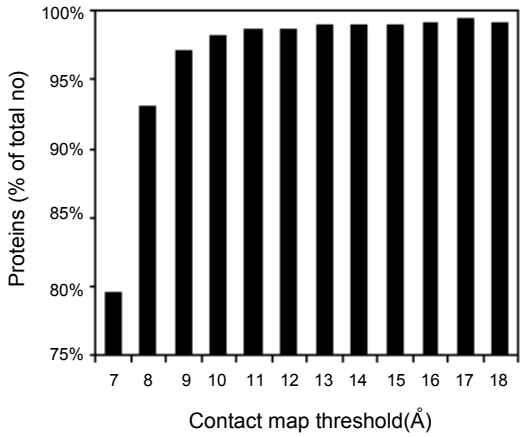


Figure 4.2: Proteins within our set for which COMAR finds a 3D structure consistent with the corresponding input contact map computed at different threshold values without random perturbations

COMAR capability of finding a 3D structure with a given input contact map depends therefore on the interplay between the quality of the initial guessed solution (Phase 1) and the results after the refinement procedure (Phase 2). For proving this the robustness of the first phase and of the second phase were tested independently by evaluating the RMSD value of each 3D model to the corresponding native structure before and after the refinement. All tests have been performed on the dataset described previously, adopting a C_α representation and computing the contact map with threshold value of 12 Å. Such choice is consistent with the observation that contact maps computed at lower thresholds are found to admit 3D structures which, in spite of being completely consistent with the input contact maps, are largely different from the native structures. It is interesting to discuss how much the random perturbation of our statistical information is relevant in order to obtain convergent computations. We tested what is the percentage of convergence of COMAR when the initial solution is not randomly perturbed. For instance, COMAR converges for 98.69% for contact maps of threshold 12 Å.

In Figure 4.2 the percentage of convergence of COMAR for different values of contact map threshold values is reported. We obtain that the convergence rate is above 90% for all thresholds over 7 Å. The reconstruction quality is higher for thresholds ranging between 10 and 18 Å.

4.4.4 Quality of the prediction phase

The RANDOM-PREDICT procedure tries to guess a possible set of coordinates for a given contact map by using available statistical information on contact distribution distances in real proteins. The prediction is partially random in the sense that the predicted set of distances is actually obtained by introducing random perturbations on a set of distances recovered from statistical information. The quality of the prediction phase can be measured in the terms of the RMSD from the native structure. Series of tests were performed for both the sets of distances generated with and without randomness. In Figure 4.3 it is shown how the proteins in our data-set are distributed according to the RMSD between the native structure and the non-random initial structure. The maximum RMSD value considered is the average RMSD value obtained after 50 different runs. The maximum RMSD value reached is 19.4 Åat an average RMSD of 3.1 Å.

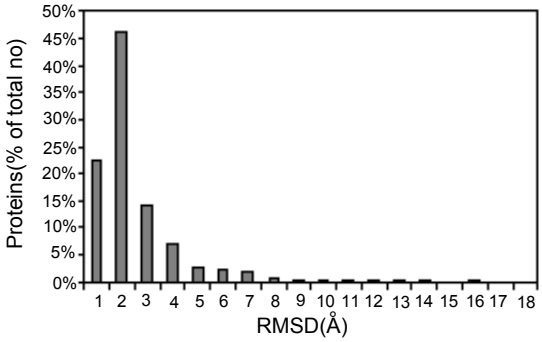


Figure 4.3: Distribution of proteins according to RMSD value between the native 3D structure and the guessed initial structure as evaluated after the RANDOM-PREDICT phase without random perturbation

In Figure 4.4 we show the results of the same test when the initial guessed solutions are randomly perturbed. For each protein the RMSD value considered is the average RMSD value obtained after 50 different runs. The maximum RMSD value reached is 25.5 Å at an average RMSD of 4.7 Å.

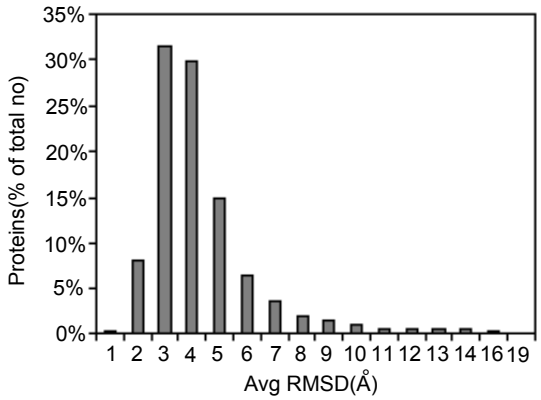


Figure 4.4: Distribution of proteins according to RMSD value between the native 3D structure and the guessed initial structure as evaluated after the RANDOM-PREDICT phase with random perturbation

From the test results, it appears that initial structures guessed without randomization have an average better quality when compared to the native ones. However, as shown in Figure 4.2, when randomization is omitted, the algorithm procedure fails in recovering all 3D models as a function of the threshold value. As a test case, the initial non-random solution of the protein of the Cricket Paralysis Virus (1b35, chain B) has very high RMSD value (19.4 Å) from the native structure. Alternatively, when randomization is introduced for the same protein, among the 50 random initial structures generated at least some have RMSD lower than 5 Åfrom the native structure. This is an example of how randomizing on the initial set of coordinates can effectively improve the performance of COMAR when the non random initial solution leads to non convergence.

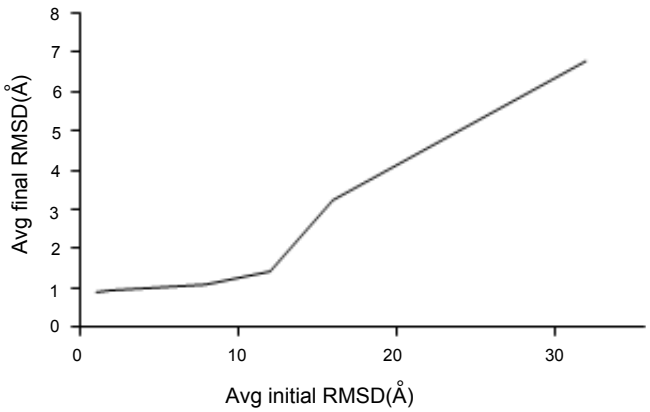


Figure 4.5: Average RMSD from the native structure of structures refined by Phase 2 of COMAR as function of the RMSD of the initial structure from the native structure. See text for details

4.4.5 Error tolerance of the refinement phase

The test of the convergence of the refinement in terms of RMSD of the initial solution to the native structures is done randomly generating structures with RMSD values ranging between 1 Åto 32 Å. A native set of coordinates C is perturbed with maximum error n Å, $n \in [1, 32]$, by randomly moving every coordinate in C of at most n Å. Experiments have shown that a random perturbation with maximum error n Ågenerates a 3D structures whose RMSD from the original one is around n Å. For each native structures a series of 10 random tests were performed. The percentage of convergence in terms of the class of errors is shown in Figure 4.6 All native structures perturbed up to 8 ÅRMSD are refined to structures matching exactly the native contact maps. The number of non converging structures is rapidly increasing when the RMSD value from the native structure is above 12 Å. This indicates that COMAR has good convergence capability: in nearly all tested cases Phase 1 generates an initial structure having RMSD which is at most 8 Åfrom the native one (see Figure 4.4).

This is further corroborated by the fact that Phase 2 can greatly reduce the RMSD of the given structure from the native one even when convergence is not

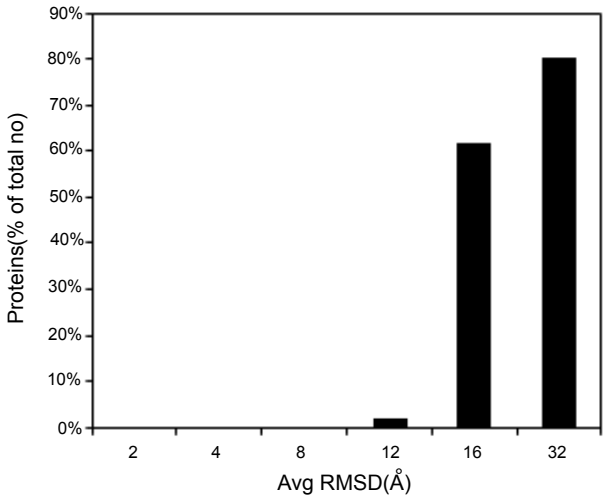
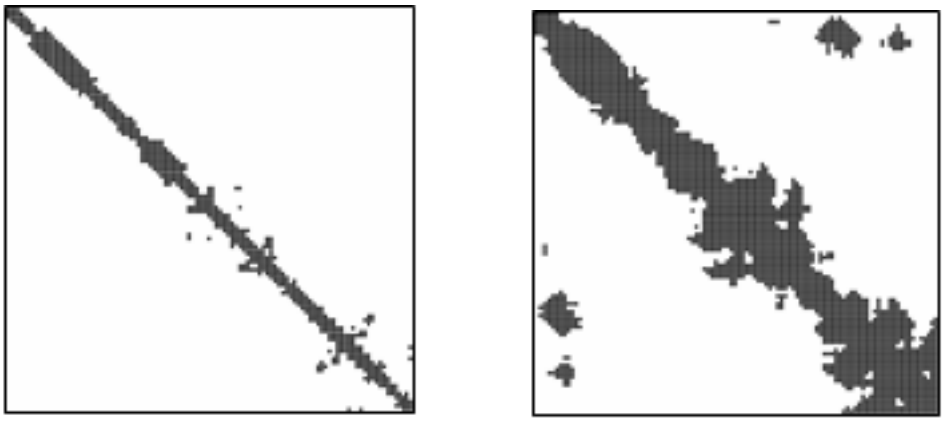


Figure 4.6: Percentage of proteins of our set for which Phase 2 of COMAR is not able to converge as function of RMSD value of the initial structure to the native structure. See text for details.

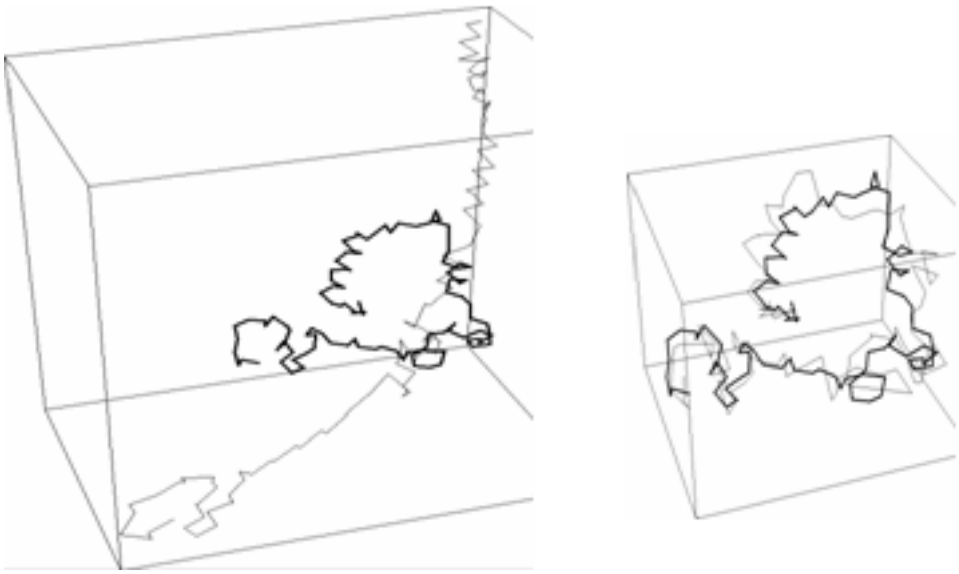
obtained (Figure 4.5). For Example, for initial structures perturbed up to 16 Å, the average RMSD obtained after the refinement procedure is 3.2 Å; 61.9% of this structures is however not consistent with the corresponding native contact map.

4.4.6 Structure Recovery

For each protein of our selected non redundant data set, containing 1760 protein structures, we generate 12 different contact maps by changing the contact threshold from 7 to 18 Å, with a 1 Åstep, and then we run our procedure for all the $12 \cdot 1760$ generated contact maps. The most relevant result of our procedure is the fact that all the reconstructed protein structures satisfy the native contact maps. This means that the Hamming distance between the native and the reconstructed contact maps is 0, or in other words, that given the contact map of a protein our algorithm finds a 3D structure which has the same contact map of the native protein. In spite of this, in some cases, the RMSD of the reconstructed protein with respect to the native structure can be very large (Table 4.1). This indicates that some contact maps can



(a) (b)



(c) (d)

Table 4.1: Contact map degeneracy: a test case. The recovery of the 3D structure of Human Myeloperoxidase Isoform C (1cxp chain B, 104 residues, all-alpha). (a) 1cxp contact map computed at a threshold of 7 Å; (b) 1cxp contact map computed at a threshold of 16 Å; (c) 1cxp native structure (thick line) compared to a recovered structure with the same contact map (a) (RMSD= 41.31Å); (d) 1cxp native structure (thick line) compared to a recovered structure with the same contact map (b) (RMSD= 4.95Å).

SCORING THE RECOVERY OF 3D STRUCTURE
FROM THE CONTACT MAPS OF 1760 PROTEINS

Threshold (Å)	Cmap dist (Å)	Avg RMSD (Å)	AvgSD RMSD (Å)	Avg Time (s)	AvgSD Time (s)
7	0	6.11	4.09	15	136
8	0	4.58	3.86	9	110
9	0	3.37	3.42	9	155
10	0	2.62	2.98	10	157
11	0	2.21	2.69	5	71
12	0	1.97	2.51	3	15
13	0	1.75	2.29	2	13
14	0	1.58	2.09	3	16
15	0	1.47	2.01	10	274
16	0	1.39	1.90	2	9
17	0	1.36	1.75	5	94
18	0	1.35	1.79	3	17

Table 4.2: Threshold = the threshold used to compute the input contact map; Cmap dist = the Hamming distance between the contact map of the native structure and the contact map of the recovered structure; Avg RMSD = the average, over all proteins, RMSD between the native structure and the recovered structure; AvgSD = the average standard deviation over all proteins; Avg Time = the average, over all proteins, time needed to recover the 3D structure.

represent a huge ensemble of protein conformations. Usually this means that the map contains only a broad central band of local contacts, and no constraints are posed on the global bending of the protein. The reconstruction ambiguity is more evident when the contact map is generated using low values of contact thresholds (ranging from 7 to 9 Å) and decreases as the contact threshold increases (Table 1). Our results indicate that at increasing contact map threshold both average RMSD and standard deviation values decreases over the all protein set (Table 4.2). At increasing threshold value global features in the contact map help in finding the 3D structure likely to be more similar/close to the native one.

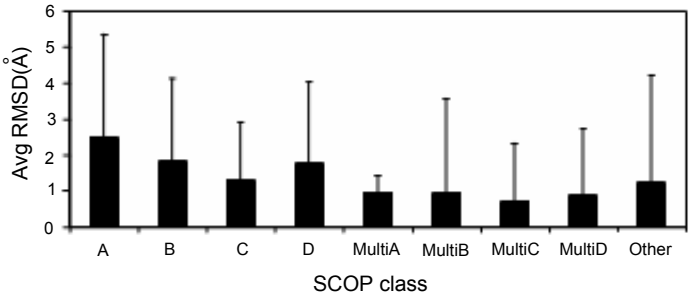


Figure 4.7: Average RMSD values on the different SCOP classes as obtained using contact maps computed with a threshold of 13 Å

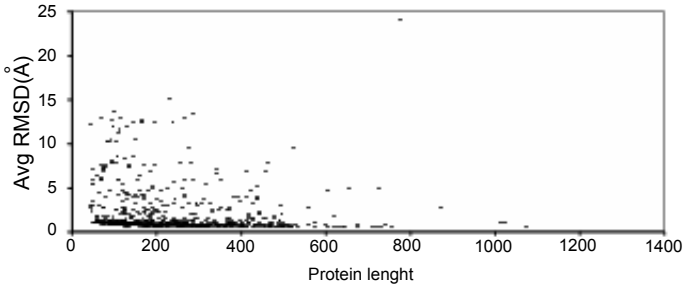


Figure 4.8: Actual RMSD distribution as function of the protein length (no of residues) when contact maps are computed with a contact threshold of 13 Å.

A typical example is shown in Table 4.1 for the protein Human Myeloperoxidase Isoform C (1cxp, chain B). The contact map computed with a threshold equal to 7

Å(Table 4.1 (a)) does not contain enough global information of the protein structure and a large number of protein structures are represented by that map. For instance, a possible reconstruction is reported in table 4.1 (c) where the RMSD to the native structure is 41.3 Å. When the contact map is computed at a threshold of 16 Å(Table 4.1 (a)) more features are available off of the main diagonal and the recovered 3D structure is closer to the native one. Indeed RMSD decreases now to 4.9 Å(Table 4.1 (d)). This finding prompted us to do a search in the threshold space to optimize the RMSD values. We find that a better 3D reconstruction is obtained when a high threshold value is adopted (10 Åor higher), while the average running time (over 1760 proteins) does not depend on the threshold adopted (table 4.2). RMSD values between the reconstructed and the corresponding native 3D protein structures are analyzed as function of the four main SCOP classes, clustered in mono and multi-domain proteins. The results are shown in Figure 4.7. As a general trend we find that multi-domain proteins are more easily reconstructed with our procedure than mono domain proteins. This is so rather independently of the threshold value adopted. One possible explanation is that the contact map of multi-domain proteins carries information about the inter-domain residue contacts that poses more constraints to the reconstruction of the 3D protein structure. Another interesting point that emerges from Figure 4.7 is the fact that the contact maps of mono-domain all-alpha proteins (A SCOP label) tend, on average, to be more ambiguous in their reconstruction. This is in agreement with the fact that all-alpha proteins are characterized by contact maps with a great number of contacts made by sequence nearest-neighbor residues and this hampers global 3D reconstruction. An analysis of our procedure as a function of the protein length shows that the method works independently of the protein size and that long proteins are on average reconstructed as well as short ones Figure 4.8.

4.4.7 Comparison with Previous Methods

To our knowledge only four methods have been introduced so far to reconstruct the protein 3D structures starting from the contact map information [5, 25, 7, 18].

The approach developed by Vendruscolo et al. [18] was tested on some 20 proteins. Unlike our results, their findings indicate that RMSD on average increases when the protein length increases. This effect may be due to the adopted simulated annealing procedure that require more optimization steps for large than for short proteins; furthermore they stop the search without a complete satisfaction of the contact maps (Cmap distance = 0). On the contrary, our method runs till the satisfaction of the contact map (table 4.2).

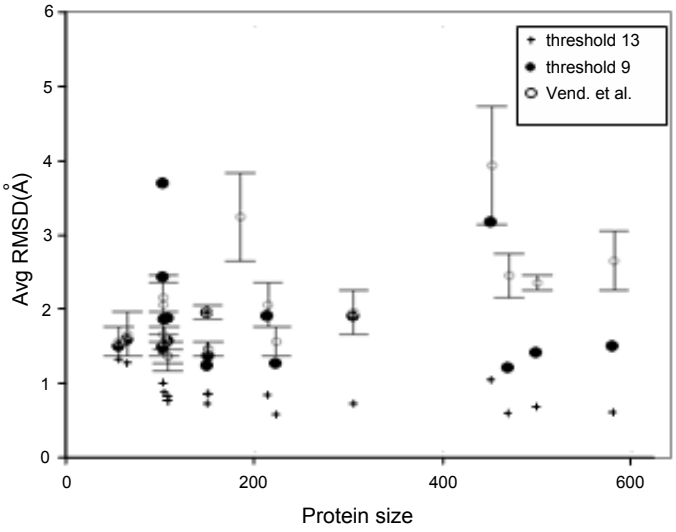


Figure 4.9: Reconstruction accuracy (RMSD) of our method on the set of Vendruscolo et al, [20]. The results correspond to a contact threshold of 9 Å (for a direct comparison with [18]) and of 13 Å, respectively. The error associated with the Vendruscolo et al. reported data is due to the fact that the complete satisfaction of the contact map is not a constraint for their search.

COMPARISON OF OUR METHOD WITH THAT OF GALAKTIONOV AND MARSHALL

PDB code	Size	Galationov Marshall RMSD(Å)	Our method RMSD(Å)	Our method Time(s)
1rdg	52	0.66	1.08	0.01
1pcy	99	0.88	0.90	0.08
4fd1	106	0.86	0.74	0.14
1acx	108	0.96	0.83	0.13
1cpv	108	0.89	0.80	0.12
Avg		0.85	0.87	0.096

Table 4.3: The protein set is the same of Galaktinov and Marshall [25]. (*) In this specific case we used a cut-off threshold of 13 Å; the results with other thresholds are similar.

Chapter 5

FT-COMAR

5.1 Adding Fault Tolerance

To test the reliability of the reconstruction performed by COMAR the algorithm stops the execution after the first run of the main cycle, i.e.: the while loop is executed just once. Such modification is necessary since a faulty contact map can not be physical, i.e. the while loop is executed just once. Such modification is necessary since a fault contact map can not be physical, i.e. there are no three-dimensional structures consistent with it and the termination condition of COMAR imposes the procedure to run forever when applied to a non-physical contact map.

FT-COMAR($CM \in \{0,1\}^{n \times n}, t \in N$)

1 $CM' \leftarrow$ FILTER(CM)

▷ Phase 1: Initial Solution

2 $C \leftarrow$ FT-RANDOM-PREDICT(CM',t)

▷ Phase 2: Refinement

3 $C \leftarrow$ FT-Correct(CM',C,t)

4 set ε to strictly positive value

5 **while** C is not consistent with CM' and $\varepsilon > 0$

6 **do** $C \leftarrow$ FT-Perturbate(CM',C,t,ε)

7 $C \leftarrow$ FT-Correct(CM',C,t)

8 decrement slightly ε

9 **return** C

FT-COMAR can work on incomplete contact maps with some unknown entries. Indeed FT-RANDOM-PREDICT, FT-Correct, FT-Perturbate are simple modifications of RANDOM-PREDICT, Correct, Perturbate, which do not consider unknown entries during the processing. Moreover, to deal with blurred contact maps, the reconstruction phase of FT-COMAR is preceded by a preprocessing of the contact map (FILTER) in order to detect (and then mark as unknown) unsafe entries of the contact map. FT-COMAR is general enough to accept any type of filtering procedure. In this work we analyze the performances of FT-COMAR in the hypothesis of a perfect FILTER, i.e. able to detect and mark as unknown exactly all faulty entries of the contact map, and with a simple filtering algorithm.

5.2 Experimental Results

We selected the proteins from SCOP release 1.67 with X-ray protein structures from the PDB, with resolution < 2.5 Å, without missed internal residues. We removed sequence redundancies using BLAST, ending up with a data-sets of 1760 protein chains with sequence similarity lower than 25%. Among these we selected 120 proteins, distributed (not uniformly) between lengths of 50 and 1100 residues.

To avoid contact maps for which we know there are very different possible structures consistent with them [16] we choose proteins whose three-dimensional structure can be reconstructed by COMAR up to a 1 Å RMSD distance from the native structure. Distribution of the resulting protein set according to the SCOP structural classes is: 8 all Alpha; 20 all Beta; 58 Alpha/Beta; 14 Alpha+Beta in the mono-domain; 3 Multi-B,C,D and 17 other consist of multi-domain proteins, for a total of 100 proteins in the mono-domain and 20 proteins in the multi-domain. The complete list is available at the URL http://vassura.web.cs.unibo.it/protlist120.tgz.

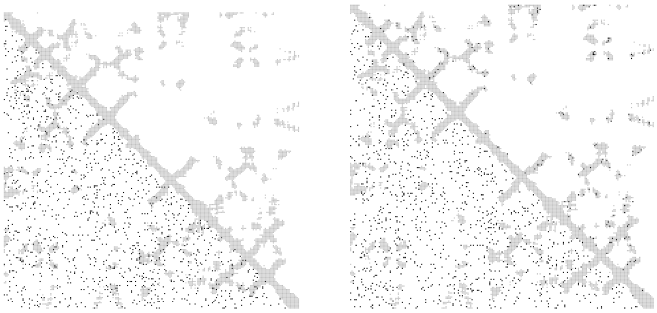
5.3 Error generation and tests configuration

To study how protein 3D structure can be reconstructed with our algorithm from faulty contact maps we introduce three classes of random errors:

- **Err.** Errors are generated by flipping the entry of randomly chosen rows and columns of the contact map. To introduce $x\%$ errors we generate x errors for each 100 couples of residues, that is $\frac{x}{100}\frac{n(n-1)}{2}$ total errors.
- **Err-0** (designed to preserve contacts). Errors are generated as before but the entry of the contact map is flipped only if it is not a contact. Here $x\%$ errors means a number of $(\frac{x}{100}\frac{n(n-1)}{2} - \#contacts)$ total errors.

We never introduce errors on the main diagonal.

In our testing, for each protein contact map and for each percentage of error considered, we generate 100 different faulty contacts maps. Thus, having 120 proteins in our set, we do 12000 tests for each percentage of error. By this, our test results have to be always considered as the average values obtained from the 100 different instances we generate. All test runs have been executed on personal computers equipped with the Intel Pentium 4 processor with clock rate of 2.8GHz and 1Gb of RAM memory. Times reported are Unix user CPU times, and are measured using the time() C library function. The Heuristic is freely available for testing on the web at the following URL: `http://vassura.web.cs.unibo.it/cmap23derr/`.



(a) (b)

Table 5.1: Contact map of a Asn102 mutant of trypsin (PDB code: 1trmA). The contact map is computed with a threshold of 12 Å: gray areas are contacts, white areas are non-contact and black areas are errors. (a) Above diagonal: native map, 24753 pairs of residues, 3595 contacts, 21158 non-contacts, and no errors. (a) Below diagonal: Err 5%, so to say (5% of 24753 =) 1237 random errors. (b) Above diagonal: Err-1 5%, that is (5% of 3595 =) 179 random errors on contacts. (b) Below diagonal: Err-0 5%, that is (5% of 21158 =) 1057 random errors on non-contacts.

5.4 Structure reconstruction from faulty contact maps

In this section we show experimental results on the behavior of COMAR with faulty contact maps. We perform tests by introducing from 1% up to 10% random errors of class Err. The average RMSD of the reconstruction from those faulty contact maps is shown in Figure 5.1. The results indicate that the quality of the protein 3D structure reconstruction depends on the protein size: proteins with less than 150 residues are reconstructed with a RMSD (from the native structure) that is less than 5 Å even when 10% random errors are introduced. For proteins with a number of residues ranging between 150 and 400, the quality of the reconstruction decreases with the increase of errors but the average RMSD still remains less than 5 Å for small percentages of errors. For proteins with more than 400 residues our algorithm shows poor performances (RMSD>5Å) even for small percentages of errors including 1% errors. Note that the sheer number of errors relative to the same percentage increases with size: as an example 10% random errors for a protein of size 100 means 450

errors, while 1% random errors for a protein of size 400 means 798 errors.

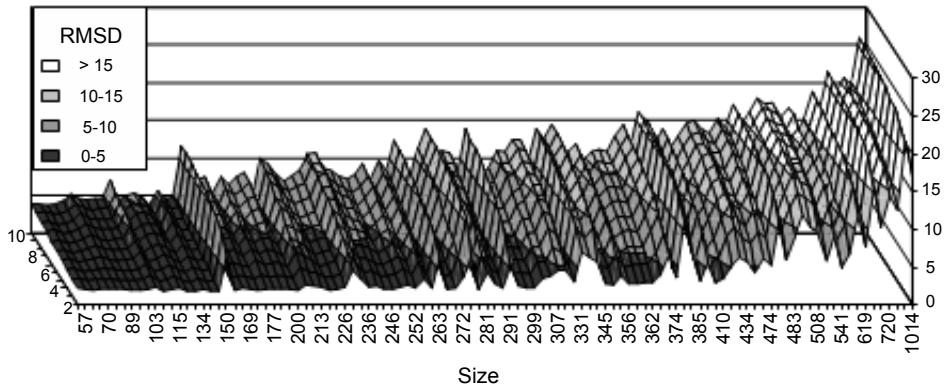


Figure 5.1: Reconstruction quality (RMSD) as function of the number of residues in the protein (Size) and of the percentage of random errors on the total pairs of residues (Err%). Better reconstruction has darker colors. As expected, reconstruction quality decreases for bigger proteins and higher percentages of errors. Note that the sheer number of errors relative to the same percentage increases with size: 10% random errors for a protein of size 100 means 450 errors, while 1% random errors for a protein of size 400 means 798 errors.

We analyze how the reconstruction quality varies among SCOP categories with the aim of highlighting whether some categories can be reconstructed better than others. In Figure 5.2 we show how reconstruction quality varies for different SCOP categories when we introduce 5% random errors. As shown in Figure 5.1, the mean RMSD from the native structure increases proportionally to protein size, with some exceptions. The most notable exception is the CDK4/6 inhibitory protein p18INK4c (1ihb chain A; (size 156) that is in the SCOP Alpha+Beta category. It appears (Figure 5.2) that exceptions to the length dependent behavior of the quality of the reconstruction are rare and distributed among SCOP categories so that it cannot be concluded that one SCOP category is more difficult to be reconstructed from faulty contact maps than another. We analyze how different types of errors influence the quality of reconstruction. In particular, in Figure 4, we compare the performance of COMAR on the three classes of errors Err, Err-0 (errors on non-contacts), Err-1 (errors on contacts). As shown in Figure 5.3, on the average, for COMAR is better to deal with Err-1 errors than with **Err-0** errors. For example, we can see that

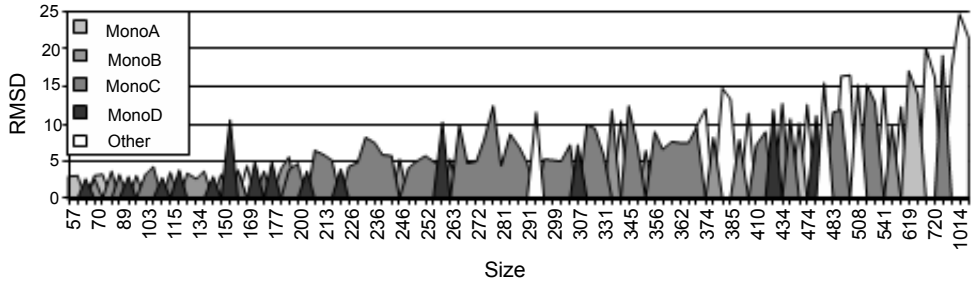


Figure 5.2: Reconstruction quality (RMSD) with an error Err 5% as a function of the protein length (Size) clustered according to SCOP categories. As expected the quality is better for small mono domain proteins, with few exceptions. Note that the exceptions do not belong to the same SCOP category, so that no category is better reconstructed with COMAR than others.

contact maps with 50% errors on contacts are reconstructed with the same quality of contact maps having 1% errors on non-contacts (which means about 10% extra contacts).

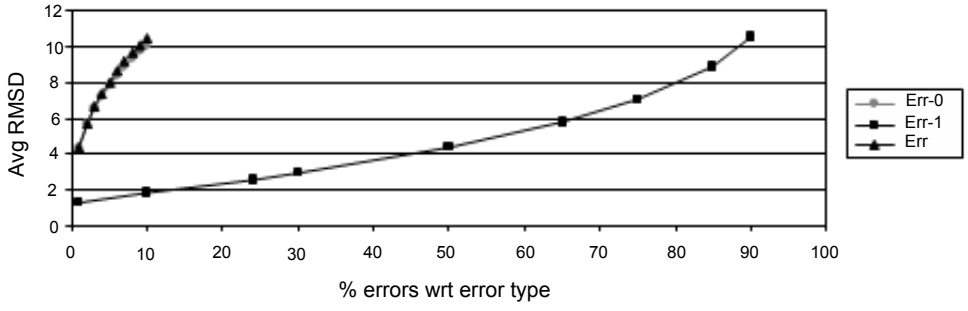


Figure 5.3: Average RMSD to the native structure of structures reconstructed from contact maps as a function of the percentage of errors with respect to (wrt) each error class: Err refers to random errors, Err-1 refers to errors on contacts and Err-0 refers to errors on non-contact. Note that reconstruction quality is better in presence of Err-1 errors.

5.5 Improving the reconstruction from faulty contact maps

Our tests give some clues on how the quality of the prediction of contact maps could influence the reconstruction phase. This is much more evident if we analyze the re-construction quality of FT-COMAR on faulty contact maps assuming to have a perfect filtering procedure, i.e. a procedure which is able to detect all errors on faulty contact maps. To test this approach we generate random incomplete contact maps by randomly choosing a column and a row of the contact map and marking that entry, corresponding to a detected error, as not safe (to be not considered during the reconstruction routine). As shown in Figure 5.4, FT-COMAR with perfect filtering can skip up to 75% of the contact map area and still compute a reconstructed 3D structure which is endowed with a RMSD < 4 Å from the native structure. Furthermore this reconstruction quality is independent of the protein size. This unexpected result is due to the fact that FT-COMAR does not consider skipped entries in the refinement phase . In this way FT-COMAR do not uses wrong information during the refinement phase avoiding the propagations of errors. The drawback is that this is true only assuming that the remaining entries of the contact map are correct, i.e. only in presence of a perfect filtering. As shown in Figure 5.5, even if we skip only 25% of the entries, the reconstruction quality is rapidly decreasing decreasing at the increasing of errors on the remaining 75% of the map. Again note that in this case the reconstruction quality depends on the length of the protein. We can interpret these results as an evidence of the fact that the quality of the reconstruction is negatively influenced by the erroneous predictions of some contacts more than by ignoring a consistent subset of contacts during the reconstruction.

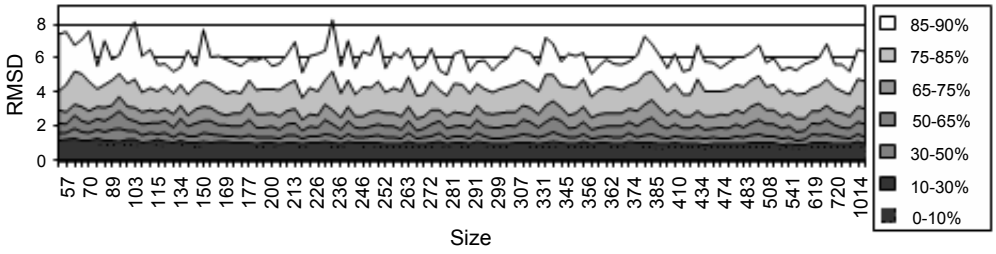


Figure 5.4: Reconstruction quality (RMSD) as function of the number of residues in the protein chain (Size) and of the percentage of random skipped pairs on the total pairs of residues (Skip%). Lower percentages of Skip have darker colors: note that we reconstruct with RMSD < 4 Å up to 75% unknown entries of the contact map for proteins of any size

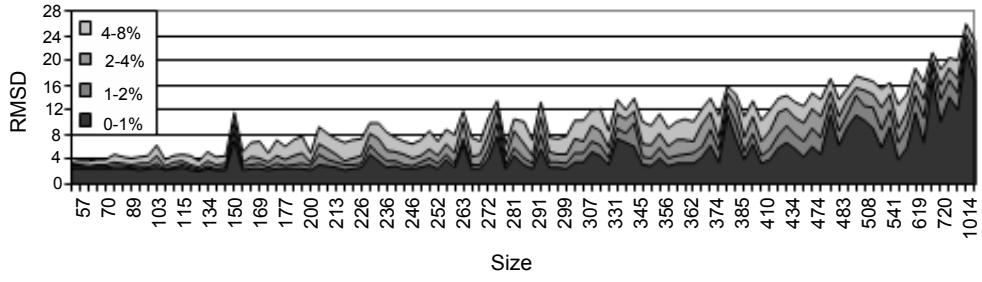


Figure 5.5: Reconstruction quality (RMSD) as function of the number of residues in the protein (Size) when 25% of the input contact map is skipped (Skip 25%). Increasing percentages of random errors (Err%) on the remaining 75% of the map are shown. Lower percentages of Err have darker colors: note that we reconstruct with RMSD < 4 Å only for low percentages of errors and reconstruction quality is decreasing at increasing protein sizes.

5.6 Error filters preprocessing with FT-COMAR

The experimental results show that we can reconstruct with much more reliability the 3D structure of a protein if we are able to predict which areas of the contact map are unsafe. This suggests that prediction quality is more important than quantity of contacts predicted: for instance, comparing Figure 5.1 and Figure 5.4 it is evident that it is better to predict 25% of the contact map with no errors than 100% of the contact map with 5% errors. This holds especially for proteins with a high number of residues. At the present time there is no way to predict contact maps with high reliability while to check unsafe contact map areas seems to be a much more simple problem. There are various properties that can be used to test the safeness of contact map areas, from physical constraints to graph properties. Here we propose a simple filtering procedure based on the so called second connectivity property, namely the number of common contacts of two contact nodes in the undirected graph (contact map) and we analyze how this procedure improves the prediction of our algorithm on faulty contact maps. The second connectivity property roughly assumes that two residues i,j are in contact if and only if they share a high number of neighbors, i.e. there is a high number of residues which are close to both i and j. Experimentally, in our data-set of 1760 non-redundant protein chains only the 6% of residues which are in contact share less than 10 neighbors and just the 0.7% of residues which are not in contact share more that 18 neighbors. Thus our second connectivity filtering procedure skips contact i,j if:

- $C[i,j] = 1$ (i e j are in contact) and i, j share less than 10 neighbors, i.e. residue i is in contact with less than 10 residues which are in contacts also with residue j;
- $C[i,j] = 0$ (i e j are not in contact) and i, j share more than 18 neighbors, i.e. residue i is in contact with more than 18 residues which are in contacts also with residue j.

Results for reconstruction quality using FT-COMAR with the simple filter described above are shown in Figure 5.6. We note that for percentages of errors less than 8%

the reconstruction quality is independent from the protein length, as in Figure 5.4. This means that the filter skips large enough faulty areas to avoid their negative influence on the whole reconstruction. When errors are over 16% the reconstruction quality decreases at the increasing of protein length. To avoid this behavior a better adjustment of filtering parameters, for example based on number of expected contacts, or another type of filtering procedure should be used. Nevertheless, in general the overall reconstruction accuracy with this simple/basic filter is significantly improved, as can it clearly seen by comparing Figure 5.1 and Figure 5.6. We remark also that our algorithms runs within minutes, allowing them to be used also for a large-scale number of predictions. The reconstruction times of FT-COMAR for our 120 proteins data set are shown in Figure 5.7.

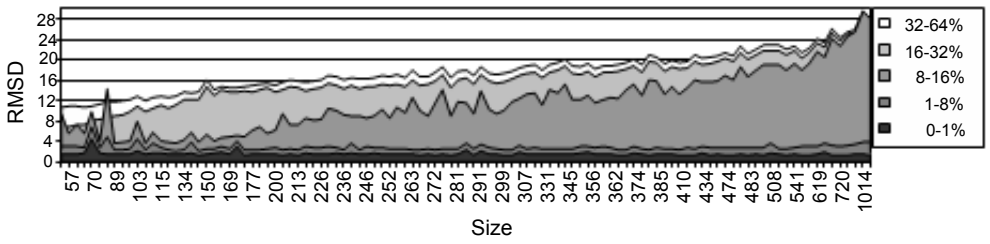


Figure 5.6: Reconstruction quality (RMSD) of FT-COMAR as function of the number of residues in the protein (Size). Lower percentages of random errors (Err%) on the whole contact map are shown with darker colors. Note that we reconstruct with RMSD < 4 Å for $1 - -8\%$ of errors for proteins of any size, while over 16% of errors the simple filtering preprocessing adopted is not able to skip enough errors to keep reconstruction quality independent from protein size.

5.7 Comparisons with previous works

In Figure 5.8 our target is the protein 1trm chain A to compare with the previous state-of-the-art reconstructing algorithm of Vendruscolo et al. [17]. The reconstruction quality is shown as a function of the number of included random errors. Both with COMAR and FT-COMAR we obtain better reconstruction quality. To compare this result with the other tests described in this work, it should be considered

that 1000 errors are approximately 4% of the total number of contact residue pairs and 4000 errors are approximately 16% of contact residue pairs.

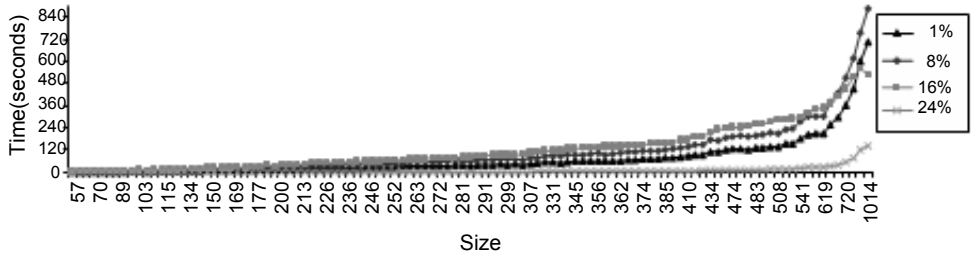


Figure 5.7: Average FT-COMAR reconstruction times in seconds for our 120 proteins data set as function of the protein length for four percentages of random errors: 1%, 8%, 16% and 64%. Note that for 64% errors the execution time of FC-COMAR decreases. In this case the quality of the reconstruction also decreases (Figure 5.6).

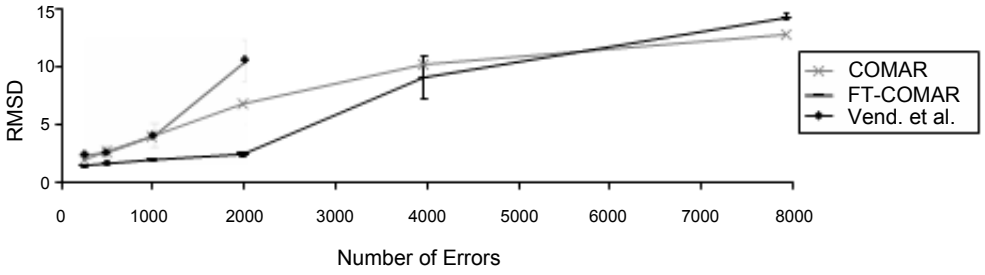


Figure 5.8: Average reconstruction accuracy (RMSD) for protein 1trm (chain A, 223 residues) as function of the number of random errors included in the native contact map. Vend refers to the performances described in [13]. Consider that 1000 errors are approximately 4% of the number of pairs of residues

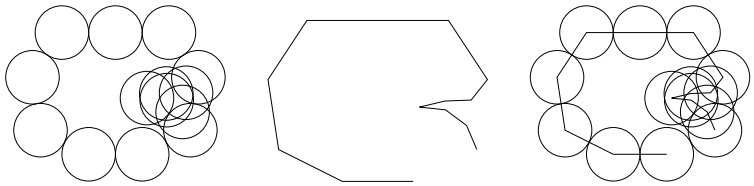
5.8 Final Considerations

We perform extensive tests of the reconstruction quality of COMAR on a set of 120 non-redundant protein chains and compared the reconstruction performances in terms of RMSD on three classes of different errors: general errors, errors on contacts (that is errors on 1-entries of contact maps) and errors on non-contacts

(that is errors on 0-entries of contact maps). The experimental results show that the reconstruction quality of contact maps with 50% errors on contacts is comparable to the reconstruction quality of contact maps with 1% errors on non-contacts. That is, COMAR is much more tolerant to errors on contacts than to errors on non-contacts. FT-COMAR can work on incomplete contact maps, i.e. contact maps with unknown entries. We showed that FT-COMAR can ignore up to 75% of the contact map and still recover a three dimensional structure from the remaining 25% entries with a RMSD value from the native one of less then 4 Å. Our conclusion is therefore that in order is to improve structure reconstruction from contact maps more emphasis should be put on the quality than on the quantity of contact predictions. This is corroborated also by the better results obtained when a simple basic filter is implemented to detect unsafe (randomly perturbed) contact map areas. The very basic filtering algorithm we develop is based on the contact second connectivity property and its performance is tested versus the reconstruction quality obtained with the not filtered faulty contact maps. The reconstruction accuracy of FT-COMAR with this simple filtering procedure is overall better and, furthermore, it results to be independent of the length of the protein for percentage of errors less than 8%. We think that on this line other more complex filtering procedures further will improve the reconstruction task.

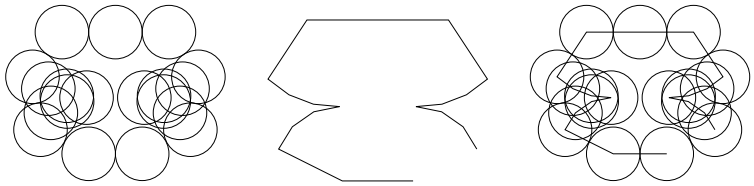
A Appendix

In the following figures, the different configurations of 2-cages we used for modeling the Grid Drawing Component are shown.



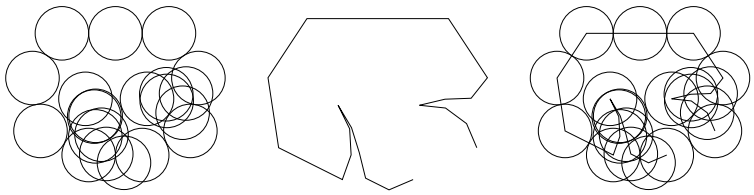
(a) (b) (c)

Table 5.2: 2-Cage. One Bead of capacity 1.



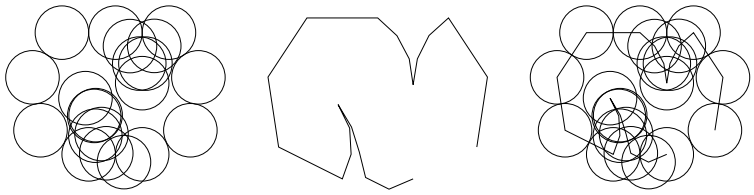
(a) (b) (c)

Table 5.3: 2-Cage. Two beads of capacity 1.(left-right)



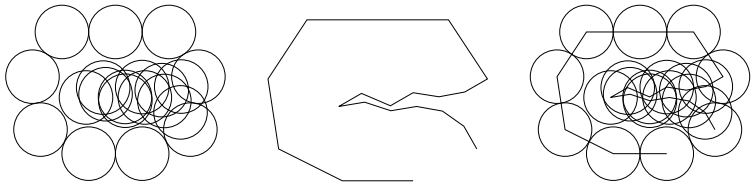
(a) (b) (c)

Table 5.4: 2-Cage. Two beads of capacity 1.(left-bottom)



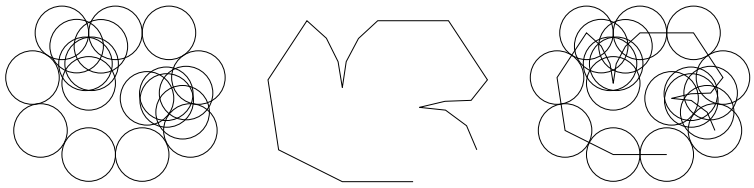
(a) (b) (c)

Table 5.5: 2-Cage. Two beads of capacity 1.(top-bottom)



(a) (b) (c)

Table 5.6: 2-Cage. One Bead of capacity 2



(a) (b) (c)

Table 5.7: 2-Cage. Two beads of capacity 1.(top-right)

Chapter 6

Conclusions

In this thesis we have faced the problem of protein reconstruction from contact map from a theoretical and experimental point of view. In the first part of this thesis we have investigated the computational complexity of the protein reconstruction problem and prove that the 2-dimensional realization problem remains NP-hard even with the backbone constraint. In the second part of the thesis we have presented COMAR, an heuristic algorithm for the reconstruction of protein 3D-structure from contact map. Performance analysis of the algorithm has shown that there exist native contact maps for which there are numerous different possible structures consistent with them. We have then evaluated the fault tolerance of COMAR introducing three different class of random errors. The analysis has shown that in general the reconstruction quality decrease with the length of protein and that the algorithm tolerates error on contact. We have then introduced an improved version of the algorithm, called FT-COMAR (fault tolerant COMAR), which experimental results show that it can ignore up to 75% of the contact map and still obtain a protein three-dimensional structures whose RMSD for the native one is less then 4 Armstrong. Furthermore the reconstruction quality is independent from protein length, which suggest that, to improve protein reconstruction from contact maps, contact map prediction should put more emphasis on prediction quality instead of quantity.

References

[1] L.M. Blumental. *Theory and applications of distance geometry.* 1970.

[2] J.M. Chandonia and S.E. Brenner. The impact of structural genomics: Expectations and Outcomes. *Science*, (311):347–351, 2006.

[3] Deegan R. Cook. The Alta Summit. *Genomics.*, (5):661–663, 1984.

[4] Charles DeLisi. Genomes: 15 Years Later A Perspective by Charles DeLisi, HGP Pioneer. *Human Genome News*, (11), 2001.

[5] J. Bohr et al. Protein structures from distance inequalities. *J. Mol. Biol.*, (231):861–869, 1993.

[6] Fabian Kuhn,Thomas Moscibroda and Roger Wattenhofer. Unit Disk Graph Approximation. *Workshop on Discrete Algothrithms and Methods for MOBILE Computing and Communications*, pages 17–23, 2004.

[7] P. Frasconi P. Baldi. G. Pollastri, A. Vullo. Modular DAG-RNN Architectures for Assembling Coarse Protein Structures. *J. Comp. Biol.*, (13:3):631–650, 2006.

[8] T.F. Havel. G.M. Crippen. *Distance geometry and molecular conformation.* 1988.

[9] D.G. Kirkpatrick H. Breu. Unit disk graph recognition is NP-hard. *Computational Geometry*, (9):3–24, 1998.

[10] T.F. Havel. *Distance Geometry: Theory, Algorithms, and Chemical Applications.* 1998.

[11] Z. Wu. J. Moré. ε-optimal solutions to distance geometry problems via global continuation. *P. M. Pardalos, D. Shalloway, and G. Xue, editors, Global Minimization of Non-convex Energy Functions: Molecular Conformation and Protein Folding*, pages 151–168, 1995.

[12] K.S. Booth and G.S. Luecker. Testing for consecutives ones property,interval graph and graph planarity using PQ-tree algorithms. *J. Computer System Science*, (13):335–379, 1971.

[13] P. Fariselli P.L. Martelli R. Casadio. L. Bartoli, E. Capriotti. The pros and cons of predicting protein contact maps. *Protein Structure Prediction: Methods and Protocols Humana Press (in press).*

[14] Ng PC Feuk L Halpern AL et al. Levy S, Sutton G. The Diploid Genome Sequence of an Individual Human. *PLoS Biology*, (5), 2007.

[15] P. Di Lena F. Medri P. Fariselli R. Casadio M. Vassura, L. Margara. FT-COMAR: fault tolerant three-dimensional structure reconstruction from protein contact maps. *Bioinformatics*, 2008.

[16] P. Di Lena F. Medri P. Fariselli R. Casadio. M. Vassura, L. Margara. Reconstruction of 3D Structures From Protein Contact Maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(3):578–589, 2008.

[17] E. Domany. M. Vendruscolo. Protein folding using contact maps. *Vitam Horm*, (58):171–212, 2000.

[18] E. Kussell M. Vendruscolo and E. Domany. Recovery of protein structure from contact maps. *Folding and Design*, (2(5)):295– 306, 1997.

[19] D.S. Johnson M.R. Garey. *Computers and Intractability: A guide to the Theory of NP-Completeness.* 1979.

[20] A. Valencia R. Casadio P. Fariselli, O. Olmea. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins*, (45 Suppl 5):157–162, 2001.

[21] C.A. Orengo R.L. Marsden, T.E. Lewis. Towards a comprehensive structural coverage of completed genomes: A structural genomics viewpoint. *BMC Bioinformatics*, (8), 2007.

[22] C. Sander E.E. Abola R.W.W. Hooft, G. Vriend. Errors in protein structures. *Nature*, (381), 1996.

[23] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. *J. Comp. Biol.*, pages 480–489, 1979.

[24] A.A. Schaffer J. Zhang Z. Zhang W. Miller D.J. Lipman. S.F. Altschul, T.L. Madden. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, (25(17)):3389–402, Sep 1997.

[25] G.R. Marshall. S.G. Galaktinov. Properties of intraglobular contacts in proteins: an approach to prediction of tertiary structure. *System Sciences*, (45 Suppl 5):326 – 335, Jan 1994.