STRUCTURE APPROXIMATING INVERSE PROTEIN FOLDING IN 2D AND 3D HPC MODELS

by

Alireza Hadj Khodabakhshi

B.Sc., Teacher Training University, 1999

M.Sc., Sharif University of Technology, 2001

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Fall 2008

APPROVAL

Name: Alireza Hadj Khodabakhshi

Degree: Doctor of Philosophy

Title of Thesis: Structure approximating inverse protein folding in 2D and 3D HPC models

Examining Committee: Dr. Pavol Hell
Chair

Dr. Arvind Gupta, Professor, Comp. Science
Simon Fraser University
Senior Supervisor

Dr. Ján Maňuch, Adjunct Professor, Math.
Simon Fraser University
Co-Supervisor

Dr. Ladislav Stacho, Associate Professor, Math.
Simon Fraser University
Supervisor

Dr. Ramesh Krishnamurti, Professor, Comp. Science
Simon Fraser University
SFU Examiner

Dr. David Bremner, External Examiner,
Professor, Comp. Science
University of New Brunswick

Date Approved: _____

Abstract

The inverse protein folding (IPF) problem is that of designing an amino acid sequence which folds into a prescribed conformation/structure. This problem arises in drug design where a particular structure is necessary to ensure proper protein-protein interactions.

Our goal here is to solve the structure approximating IPF problem in 2D and 3D in HP models. As for the 2D case, we consider a subclass of linear constructible structures designed by Gupta et. al 2004. These structures, called wave structures, are rich enough to approximate (although more coarsely) any given structure. We formally prove that protein sequence of any wave structure is stable under the HPC model. To prove the stability of wave structures we developed a computational tool, called 2DHPSolver, which we used to perform the large case analysis required for the proofs. 2DHPSolver can be used to prove the stability of any design in 2D square lattice.

For the 3D case we introduce a robust class of protein structures, called tubular structures for 3D hexagonal prism lattice. These structures are capable of approximating target 3D shapes. Interestingly, the main building block of tubular structures, a tube, consists of six parallel "alpha helix"-like structures. Similar designs appear in nature as a *coiled coil* structural motif in which 2–6 alpha-helices are coiled together. We show that the tubular structures are native for their proteins and we prove that a basic but infinite class of tubular structures consisting of a connector and three tubes of arbitrary length are structurally stable under the HPC model. Despite the tremendous amount of work on protein design for 2D lattices, to the best of our knowledge, this is the first general design of arbitrary long stable proteins for a 3D lattice.

To Marjan

To my parents

"Out beyond ideas of wrongdoing and rightdoing, there is a field. Ill meet you there."

— Rumi (1207-1273 AD)

Acknowledgments

I am mostly grateful to my friend and adviser Dr. Ján Maňuch for all his support and help on this work. I am greatly thankful to my supervisor Dr. Arvind Gupta for all his supports and encouragement during the course of my PhD studies. Many thanks to Dr. Arash Rafiey for his contributions and helps on this work. I am grateful to the other member of my committee Dr. Ladislav Stacho for all of his supports. I deeply thank Dr. Ramesh Krishnamurti for encouraging me to step into the beautiful world of theoretical computer science and also for his time for reviewing my thesis. I should also thank Dr. David Bremner for the time he spend on reviewing my thesis.

I truly appreciate the kindness and support of all my friends over the years. I specially thank Moslem Kazemi, Reza Ghorbani, Hamidreza Chitsaz, Bashir Sadjad, Vahab Mirrokni, Amir Forough Nasirai and my M.Sc. adviser Dr. Mansour Jamzad for helping me in the application process for the PhD program at Simon Fraser University. During the course of my PhD studies I worked on several other interesting problems in Computational Biology and I have been very fortunate to work with my wonderful friends and colleagues Mahdi Mirzazadeh and Masoud Harati.

I am deeply thankful to my parents for their unconditional love and support in every moment of my life.

I cannot thank enough Marjan the love of my life for her kindness, encouragements and endless supports in this journey. She is my source of strength and this thesis would never have been possible without her by my side.

Contents

List of Tables

List of Figures

List of Programs

Chapter 1

Introduction

1.1 Motivation

A protein is a polymer of tens to thousands of amino acids. There are 20 types of naturally existing amino acids that chain together via peptide bonds to form polypeptides molecules commonly know as proteins.

Proteins play a vital role in the activities within cells of living organisms. The protein functionality covers a broad range including but not limited to catalyzing the biochemical reactions necessary for life (the *enzyme* proteins), providing structural or mechanical functions that helps give cells integrity and shape, and providing means of communication within a cell and between cells (*hormone* proteins). Proteins can also bind and carry substances. For instance hemoglobin carries oxygen from the lungs to other parts of body and *myoglobin* stores oxygen in muscle tissue until it is used.

It has long been known that protein interactions depend on their native three-dimensional (3D) fold and understanding the folding processes and determining these folds is a long standing problem in molecular biology. Naturally occurring proteins fold so as to minimize total free energy. However, it is not known how a protein can choose the minimum energy fold amongst all possible folds [42]. Many forces act on the protein which contribute to changes in free energy including hydrogen bonding, van der Waals interactions, intrinsic propensities, ion pairing, disulfide bridges and hydrophobic interactions.

The native fold of proteins can be experimentally determined using X-ray crystallographic and/or nuclear magnetic resonance (NMR) methods. However, such methods are

very costly, highly labor intensive, and limited in many ways [152]. For instance, the effectiveness of the X-ray crystallographic is severely dependent on the availability of the appropriate crystals for analysis [28]. Such crystals can be difficult to produce for certain proteins (for instance, the intrinsic membrane proteins [105]). This has lead the scientists to attempt to predict the native 3D structures of the proteins using computational methods. Despite the large amount of effort expended in the prediction of protein structures during the past 30 years, this problem remains largely unsolved.

1.2 Inverse protein folding

In many applications such as drug design and nanotechnology, we are interested in the complement problem to protein folding: *inverse protein folding (IPF)* or *protein design*. The IPF problem involves starting with a prescribed target fold and designing an amino acid sequence whose native fold is the target (positive design). A major challenge in designing proteins that attain a specific native fold is to avoid proteins that have multiple native folds (negative design). The inverse and forward protein folding (protein folding prediction) are highly related, in the sense that any achievement in one will help better understand the other.

The success of the inverse and forward protein folding methods is highly dependent on the modeling of the protein folding process and choice of appropriate energy function. Such models and functions could range from very high resolution molecular dynamics models in which the Newton's laws of motion on smaller molecules are numerically solved, to the very abstract lattice models in which the placement of the amino acids is limited to the vertices of the underlying lattice model. Although the molecular dynamic models have contributed much to our understanding of the protein folding process [111], finding the minimum energy conformation of protein molecules in these models is beyond the computing power of today's computers [19]. Thus, exploring the relationships of amino acids sequences to native structures requires simplified models that average out the effects of the sequence, and atomic-resolution molecular dynamic simulations. This has led scientists to introduce simplified models in which the amino acids are represented as very simple components that can have a limited number of states in a protein conformation. Two major simplified models are *side-chain rotamer* (SCR) and hydrophobic-polar (HP) models. The great advantage of such simplified models is that we can solve the problem exactly, at least for short proteins,

by enumerating all possibilities. In SCR models only a discrete set of amino acids states and side-chain orientation relative to the target backbone structure is optimized according to a scoring function. The side-chain conformations that are considered in SCR models are the ones that most likely occur in low energy protein folds. These discrete side-chain conformations are called *rotamers*.

The HP model, introduced by Dill [41], is based on the assumption that a major contribution to the free energy of the native conformation of a protein is due to interactions between hydrophobic amino acids. In HP model the 20 amino acids from which proteins are formed are replaced by two types of monomers: hydrophobic (H or '1') or polar (P or '0') depending on their affinity to water. The hydrophobic amino acids are non-polar and thus prefer other neutral molecules and non-polar solvents. When a protein is placed in water the hydrophobic monomers tend to form clusters to minimize their contact surface with water, while the majority of polar monomers move to the surface of the protein. The HP model is often defined with respect to an underlying lattice where proteins are laid out on vertices of the lattice with each monomer occupying exactly one vertex and neighboring monomers occupying neighboring vertices. The free energy is minimized when the maximum number of pairs of hydrophobic monomers that are not consecutive in the protein sequence, are adjacent in the lattice (such pairs are called *HH contacts*). Therefore, the "native" folds are those with the maximum number of such HH contacts.

In natural proteins, sulfide bridges between two cysteine monomers play an important role in the stability of the protein structure [73]. Based on this, we extend the HP model by considering a third type of monomer, cysteines (C or '2'), and incorporating disulfide bridges between two cysteines into the energy model. We call this model the hydrophobic-polar-cysteine (HPC) model. The cysteine monomers in the HPC model are hydrophobic, but in addition two neighboring cysteines can form a sulfide-sulfide bridge to further reduce the energy of the fold. We formally prove that adding disulfide bridges to designed protein sequences indeed helps in stabilizing them.

Despite the simplicity of the HP model, the folding process in the model have behavioral similarities with the folding process of actual proteins. Although the HP model is the simplest model for protein folding, computationally it is an NP-hard problem for both the 2D [27] square and the 3D [7] cubic lattices. The hardness of the inverse protein folding under the standard definition of the HP model is still unknown but it is conjectured to also be NP-hard. Several heuristic based algorithms have been described that attempt

to solve IPF problem but none of them guarantee that the designed sequences achieve their minimum energy when fold into the target conformations (positive design criteria) and that they have unique minimum energy conformations (negative design criteria). These heuristic methods can be classified into two categories. The methods in the first category use observations about the properties of proteins to justify algorithms that design sequences [82, 166]. The second category of heuristic methods are those in which an alternative formulation of IPF, a heuristic sequence design (HSD), is considered [40, 92, 145, 155, 66, 9]. Two HSD problems, in the *canonical* and the *grand canonical* models, were introduced in [145] and [155], respectively. The HSD problems look for protein sequence with the smallest energy when folded into the target conformation. In the canonical model the number of hydrophobic monomers that can be used in a protein sequence is limited by fixing the maximum ratio between hydrophobic and hydrophilic amino acids. This condition is needed because the conformational energy can be minimized simply by using the sequence of all hydrophobic monomers, but this sequence is unlikely to achieve its lowest energy with the given target conformation. In the grand canonical model [155], the number of hydrophobic monomers is limited by adjusting the energy function instead. For instance, a simplified formulation of the grand canonical model used in [66] assumes that every hydrophobic contact contributes -2 to the total energy, every solvent accessible site of a hydrophobic amino acid contributes 1, and all other interactions do not contribute to the total energy. Since the hydrophobic monomers are penalized for their exposure to solvent, this contact potential implicitly limits the number of hydrophobic monomers in the sequence. It has been shown that the protein sequence design problem can be solved in polynomial time in the grand canonical model for both 2D and 3D square lattices, cf. [66], and in polynomial time for 2D lattices while the problem is NP-hard for 3D square lattice in the canonical model, cf. [9]. Note however, that the designed heuristic sequences under these two models are not guaranteed to satisfy the two criteria (positive and negative design) of the IPF problem.

In [60], the IPF problem was studied from a different perspective. Instead of designing a sequence directly for the target fold and relaxing conditions on the sequence the authors introduced a design method in 2D square lattice under the HP model that can approximate any target conformation and showed that approximated structures, called *constructible* structures, are native for designed proteins (positive design). They conjectured that the assigned sequences are also stable (i.e., have unique native folds) but only proved it for an infinite class of very basic structures (arbitrary long “I” and “L” shapes), as well

as computationally tested for over 48,000 structures (including all structures with up to 9 tiles). Design of stable proteins of arbitrary lengths in the HP model was also studied by [1] (for 2D square lattice) and by [101] (for 2D triangular lattice), motivated by a popular paper of Brian Hayes [68].

1.3 Our contributions

Contributions in this dissertation are two fold. First we introduce a rich subclass of *constructible* structures that is able to approximate any given shape in the 2D square lattice. We formally prove that the designed proteins are stable under the HPC model. Our result partially confirms the conjecture proposed in [60].

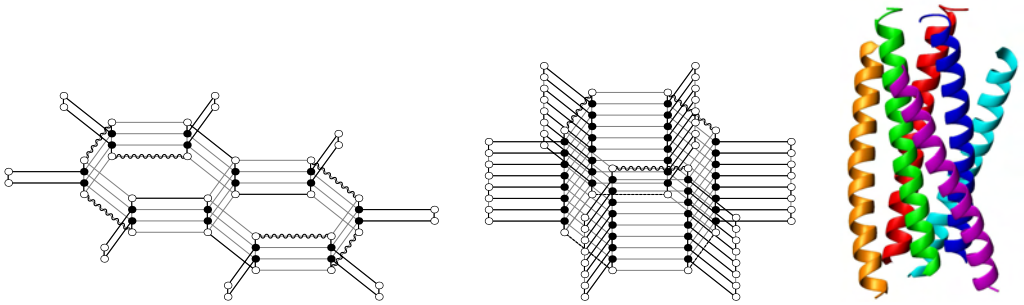


Figure 1.1: An example of (a) a connector (H and P monomers are depicted by black and white beads, respectively); (b) a tube; (c) a coiled coil structure formed by 6 alpha-helices in protein gp41. Figure (c) is taken from wikipedia (`http://en.wikipedia.org/wiki/Image:Gp41_coiled_coil_hexamer_1aik_sideview.png`) and is used by permission of WillowW.

Our second contribution is the study of the structure approximating inverse protein folding in a 3D setting. A very important consideration is the choice of the most appropriate type of 3D lattice. This question was thoroughly studied in [112]. Based on the analysis of selected protein structures from Protein Data Bank (PDB) [136] the authors found that the ideal IPF lattice should have uniform edge lengths of $3.8\mathring{A}$, minimum distance between any two vertices of $3.8\mathring{A}$, mainly 90 and 120 angles and have periodic structure. As an initial step in IPF in 3D HP model, they chose a simple lattice out of the good candidates, *the hexagonal prism lattice* and proposed a basic building tile. Based on this we first introduce an infinite class of protein structures in hexagonal prism lattice called *the tubular structures*. The building blocks of tubular structures are *tubes* and *connectors*. An example of a connector

and a tube is shown in Figure 1.1(a)-(b), respectively, where hydrophobic monomers are depicted with black beads, and polar ones with white beads. A tube consists of six parallel "alpha helix"-like structures and interestingly similar designs appear in nature as a *coiled coil* structural motif in which 2–6 alpha-helices are coiled together, cf. Figure 1.1(c). Many coiled coil type proteins are involved in important biological functions such as the regulation of gene expression, e.g., transcription factors [165, 109]. Two tubes and a tube and a connector can be connected by overlapping the bottom loop of one of the tubes and the top loop of the second tube/connector. Figure 1.2 depicts an example of a tubular structure consists of three tubes and a connector. We show that each protein of a tubular structure folds into the corresponding tubular structure, and that the proteins for the tubular structures with three tubes and a connector, arranged as in Figure 1.2, are structurally stable under the HPC model. Notice that this class contains an infinite number of protein sequences as the tubes can be arbitrary large.

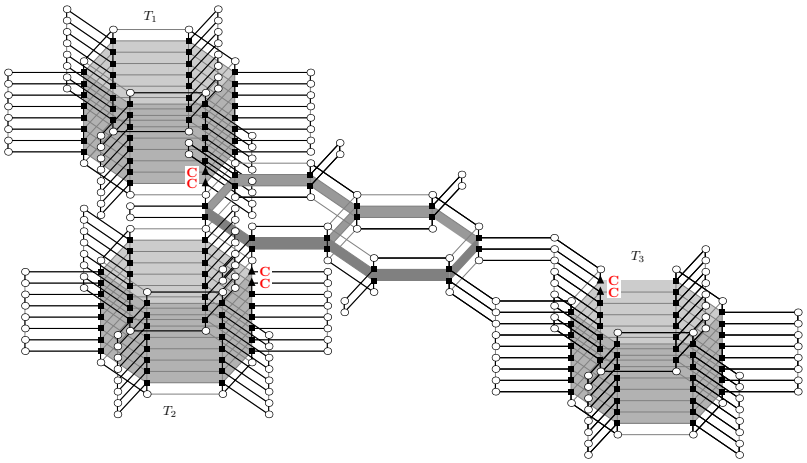


Figure 1.2: An example of a tubular structure with 3 tubes attached to the connector.

The tubular structures are sufficiently robust to roughly approximate any given shape.

For a protein for each of our structures, it is guaranteed that the designed structure is one of the native folds of the protein under HPC model. We conjecture that the proteins of our tubular structures are structurally stable, i.e., our designed proteins fold uniquely up to the structures into designed conformations. We are able to prove this formally for infinite subclasses of the simple structures (consisting of one connector and three tubes, cf. Figure 1.2). We assume that each of three tubes is sufficiently long. In addition, we assume that our proteins are closed chains of monomers, a similar assumption as used in [1], i.e.,

that the beginning and the end of the sequence are adjacent in the lattice.

1.4 Organization

The rest of this dissertation is organized as follows. In chapter 2, we present some basic background knowledge on the proteins and their structures. In chapter 3, we review some of the main models and methods for the inverse protein folding problem. In chapter 4, we introduce our structure approximating design in 2D square lattice and prove the stability of our designed proteins under the HPC model. We introduce our structure approximating design in 3D hexagonal prism lattice in chapter 5 and prove the stability of a simple but infinite subclass of the designed proteins in HPC model. Finally in chapter 6, we state some interesting open problems.

Chapter 2

Background

In this chapter we present some basic knowledge on proteins and their structures. More detail can be found in [2, 15, 162, 10, 13, 43, 49, 77].

2.1 From Sequences to Structures

Heredity is a central part of the definition of life. Each species reproduces itself faithfully by handing down information specifying, in extraordinary detail, the characteristics that the offspring shall have. Such detailed characteristics are encoded in a linear chemical code, called *DNA*, that exists in each cell of a living organism. DNA is a double-stranded polymer consisting of simple subunits called *nucleotide*. There are four distinct nucleotides labeled as A, G, C, and T.

The characteristics and the functions of a living cell are expressed by the proteins and in fact every activity carried out by living cells uses one or more proteins. Proteins, like DNA, are polymer chains of simple chemical compounds. The chemical compounds that make proteins are called *amino acids*. There are 20 natural amino acids. Therefore, one can represent the proteins as strings over a 20 letter alphabet.

The cell uses the information encoded in its DNA to synthesize proteins. Protein synthesis is a two step process: in the first step, called *transcription*, segments of the DNA sequence are used to guide the synthesis of molecules of RNA called *messenger RNA* or *mRNA*. Unlike DNA, RNA is a single stranded polymer. The building units of RNA are the same as those of DNA except that the T unit is replaced by a different monomer called U. In the second step, called *translation*, the synthesized mRNA molecules are used to guide the

synthesis of molecules of protein. The information in the sequence of an mRNA molecule is read out in groups of tree nucleotides at a time: each triple nucleotides, referred to as *codon*, determines a unique amino acid in the corresponding protein sequence. Since there are 64 possible codons but only 20 amino acids, there are cases in which several codons encode the same amino acid. The DNA segments that will be eventually translated to proteins are called *genes*.

2.2 From Structures to Functions

Protein are the machinery of life, constitute most of a cell's dry mass and in fact every activity within a living cell involves one or more proteins. Some proteins act as catalysts to facilitate chemical reactions inside the cells (enzyme proteins), some carry substances including oxygen and food to designated parts of living organism, some provide means of communication between cells (such as hormones) and others provide structural and mechanical functions that gives the cell its shape and integrity. Alisa Z. Machalek, describes the importance of protein in life as follows [106]:

"If genes are the recipes for life, then proteins are the culinary result the very stuff of life. Proteins form our bodies and direct its systems. They digest our food, help us fight infections, control our body chemistry, and in general keep us and every other living organism running smoothly. But proteins that twist into the wrong shape, have missing parts, or don't make it to their job site can cause diseases that range from cystic fibrosis to cancer and Alzheimer's."

When a protein is exposed to its natural environment, it will fold into a unique three dimensional conformation known as *native fold*. The folded conformation is stabilized mainly by non-covalent interactions between different parts of the protein sequence. These interactions include: hydrogen bonding, van der Waals interactions, intrinsic propensities, ion pairing, disulfide bridges and hydrophobic interactions. Except for disulfide bridges, all of these interactions are non-covalent. The protein folds so as to minimize its free energy, therefore the native fold of a protein is sometimes called the *minimum energy conformation*. A protein can be unfolded by treatment with certain solvents. When the denaturing solvent is removed, the protein often refolds spontaneously into its original conformation. This indicates that the native fold of a protein is unique and determined by its amino acid sequence.

The native conformation of a protein greatly determines its functionality. Most biological mechanisms at the protein level are based on shape-complementarity. This means that the proteins present particular concavities and convexities that allow them to bind to each other and other molecules to carry out the designated tasks including transferring substances, catalyzing reactions, or forming complex structures to support the shape of the cells. Therefore, determining the three dimensional native conformation of proteins is a fundamental step in studying them.

The native fold of a protein can be described in terms of several different levels ranging from local structures to complex structural interactions with other protein structures. We overview these levels briefly.

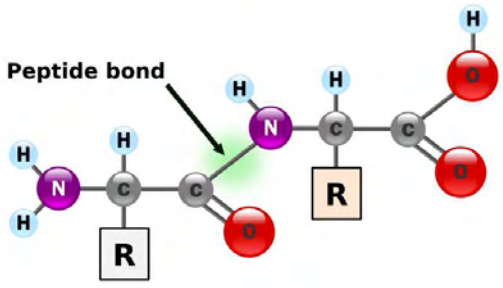


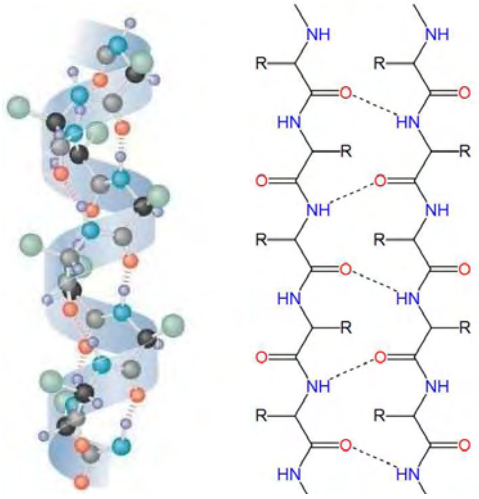
Figure 2.1: The primary structure of a protein. The sidechain groups R's, are attached to the backbone of the protein. Figure is taken from wikipedia (`http://en.wikipedia.org/wiki/Image:Peptidformationball.svg`) and is used by the permission of YassineMrabet

Primary structure

An amino acid consists of an amino group (-NH2) which is also called N-terminal, an α-carbon atom, denoted by C_α, in its center, a hydrogen atom (-H), a carboxyl group (-COOH), and a side-chain group R. The twenty standard amino acids only differ in their R group. The amino acids in a protein sequence are connected through peptide bonds. A peptide bond is formed by a chemical reaction in which a water molecule is removed and the C in the carboxyl group of one amino acid is linked directly to the N atom in the amino group of the other amino acid. Several amino acids can linearly chain together in this way to form a polypeptide (protein). The formation of a succession of peptide bonds generates a *main chain* or *backbone*, from which project the various side-chains (cf. Figure 2.1).

The primary structure of a protein describes the linear order of the amino acids contained in it. All the information needed to form the native spatial structure of a protein sequence

is encoded in its primary structure.



(a) (b)

Figure 2.2: Secondary structures of proteins: (a) alpha helix (b) beta sheet. Figures (a) and (b) are taken from wikipedia (`http://en.wikipedia.org/wiki/Image:Myoglobin.png` and `http://en.wikipedia.org/wiki/Image:Beta_sheet_bonding_parallel-color.svg`) and they are used by the permissions of National Institutes of Health and Fvasconcellos, respectively.

Secondary structures

Proteins fold up to complex structures due to the bonds formed between the side-chains. Due to the flexibility of the peptide bonds, the side-chains that are far away can also form strong bonds to contribute to the native structure of the protein sequence. The bonds that are formed between nearby side-chains along the primary sequence are called *local* or *short-range* interactions while those which are formed between side-chains that are far away are called *nonlocal* or *long-range*. The substructures of the protein's overall structure that only contain short-range interactions are called *secondary structures*. The secondary structures constitute the regular features of the protein structures. Two common types of secondary structures are α-*helix* and the β-*sheet* (cf. Figure 2.2) which can be attached together through various types of loops.

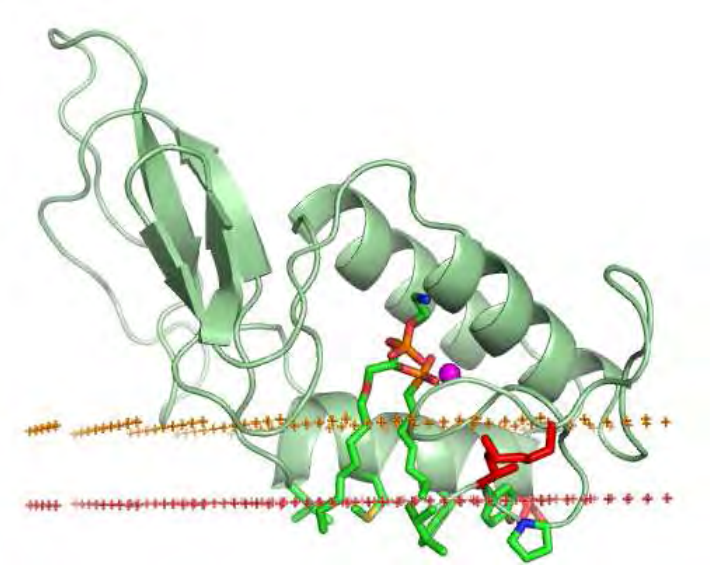


Figure 2.3: Tertiary structure of phospholipase A2 sPLA2 the bee venom. The alpha helices and beta sheets are represented by spiral and arrow ribbons, respectively. Figure is taken from wikipedia (`http://en.wikipedia.org/wiki/Image:1poc.png`) and is used by the permission of Biophys.

Tertiary structures

Due to the non-local interactions, the secondary structures in a protein can interact with each other to form a complex spatial structure called the *tertiary* structure or the *native fold*. The tertiary structure is the overall shape of a single protein molecule that is formed by the spatial relationship of the secondary structures to one another. It describes the way in which the elements of the protein's secondary structure are arranged in space (cf. Figure 2.3).

The *protein folding* problem is studying the process of folding a protein into its native tertiary structure.

Quaternary structures

Several protein structures can interact to form a multi-subunit complex called *the quaternary* structure. A fairly large number of proteins has the quaternary structure. Hemoglobin, DNA polymerase, and ion channels are examples of proteins with quaternary structure.

A protein can be divided into several different *domains*. A domain is defined as a protein sequence or subsequence that can fold independently into a stable tertiary structure.

Domains are also units of function. The number of domains in a protein range from a single domain to as many as several dozen domains. The average length of a domain is approximately 140 amino acids. Domains are formed by different combinations of secondary structures and motifs. The number of such combinations is limited and some combinations are structurally favored among others.

2.3 Protein folding prediction

Knowledge of the native fold of a protein is a prerequisite for the proper understanding of its functions. In 1958 the tertiary structure of a protein, the myoglobin, was determined using x-ray crystallographic techniques for the first time and it came as a shock to those who had hoped for simple, general principles of protein structure analogous to the simple and regular double helix DNA structure [10]. Since then the structure of many proteins has been determined using the experimental approaches mostly x-ray crystallography [44] and the nuclear magnetic resonance (NMR) technique [17]. The Protein Data Bank (PDB) [136] is a repository for tertiary structural data of proteins. The PDB currently contains the tertiary structure of 50,000 proteins and is estimated that it will triple to 150,000 structures by the year 2014 [136].

Unfortunately, the determination rate of protein structures using the experimental techniques is much slower than the rate of accumulation of amino acid sequence data [164]. This is because the experiments to determine the tertiary structures and functions are much more difficult and time-consuming than the experiments to determine the primary structures of the proteins. Over five million protein sequences have been determined [123] so far, while only the tertiary structure of 50,000 (less than one percent) of them are known experimentally. It has been estimated that there may be as many as 26 billion protein sequences in the biosphere [156] and with the anticipated rate of the sequence versus structure determination, the need for alternative methods for determining the tertiary structure is clear. The class of computational methods for predicting the structure of the proteins from their linear amino acid sequences is the most widely used alternative to experimental structure determination methods. Such computational methods are referred to as *protein folding prediction* methods. Despite the vast study on the protein folding prediction problem over the past 30 years it still remains the most challenging open problem in *proteomics* (the large-scale study of proteins, particularly their structures and functions) and perhaps computational biology.

In this section we review some of the most successful protein folding prediction methods.

The amino acid sequence of a protein contains all the information needed to determine its native conformation therefore, it is feasible to predict the tertiary structure of a protein from its amino acid sequence [6]. Most of the protein prediction approaches use a basic thermodynamic hypothesis: proteins tend to fold into a global minimum free energy state [163]. Based on this, researchers predict protein structures in two phases: first, design a scoring function to simulate the relationships between the amino acids in the native conformation, and second design efficient algorithms to find conformations that minimize the scoring function. The scoring function, sometimes referred to as *energy function*, can be constructed in two ways: from the physico-chemical principles of protein folding, or alternatively from a knowledge-based potential function, measuring the probability distribution of the possible folds of a protein.

The computational protein folding prediction methods can be divided into three categories: homology modeling, fold recognition (protein threading) and *ab initio* (new fold) methods. The first two methods are template-based while the third one predicts the protein structures without relying on any structural template.

2.3.1 Homology modeling

A set of proteins are called homologous if their genes have evolved from a common ancestral gene. Two proteins are considered homologous when they have identical amino acids in a significant number of sequential positions along the polypeptide chains [10]. The homology modeling methods rely on the fact that the structures and functions of homologous proteins are more conserved than their amino acid sequences [22, 126]. Homology modeling is widely used for protein folding prediction [108, 157, 67, 100, 65, 16, 119].

The homology modeling method consists of the following phases [163]:

- **Homologue (template) Selection:** The first step in a homology-based folding prediction method is identifying the target's homologous sequences from the structure database and determining their sequence similarity. The simplest methods for this step rely on pairwise sequence alignment of the target sequence and the database sequences. Several tools have been developed to carry out this step among which FASTA [103], BLAST [4] and PSI-BLAST [5] are used more often.

- **Sequence alignment:** In this step a multiple sequence alignment between the target

sequence and the selected templates from step one is built. The multiple alignment reveals the conserved regions in the sequences. Choosing an appropriate algorithm and scoring function are key factors in this step. For pairwise alignment, a dynamic programming algorithm such as Smith-Waterman [151] and Needleman-Wunsch [122] can be used. But if multiple templates are available no polynomial-time algorithm is known [76] therefore heuristic based methods must be used.

- **Core determination:** In this step the most conserved regions and variable segments in the sequence alignment is identified. The conserved regions are the cores of the homologous proteins and they are generally located in the secondary structures; the variable segments are treated as loops.

- **Core modeling:** The coordinates of the backbone atoms of the cores determined in the previous step are predicted by copying from those of the aligned backbone atoms of one homologous protein. A rotamer library is used to generate the coordinates of the side-chains for the cores [143, 99]. A rotamer is one of a set of conformers arising from restricted rotation about one single bond.

- **Loop modeling:** If a loop in one homologous protein is found as a suitable model for one loop in the sequence, then its coordinates from the homologous protein is copied to the target sequence; otherwise, the existing fragment database is searched to find a best fit fragment for the loop [80, 138, 154].

- **Refinement and Evaluation:** When the coordinates of the atoms of the target sequence are generated, the resulting structure is evaluated and refined. A commonly used methods is to tune the coordinates of some atoms through minimizing certain energy functions. The quality of the constructed model is also checked using some tools such as WHATIF [160] and PROCHECK [118, 95]

2.3.2 Fold Recognition (protein threading)

It is quite possible that two proteins with very different primary structures (non-homologous) have similar tertiary structures. The fold recognition, also called protein threading, is a template base method suitable for the target sequences that have similar folds as proteins of known structures but do not have homologies with them [163]. Protein threading predicts protein structure using the knowledge of the relationship between the structure and the

sequence. The basic idea behind the threading method is that the target sequence is threaded (i.e., placed and aligned) through the backbone structures of the proteins in a fold library and a *fitness* score is calculated for each sequence-structure alignment. This fitness score is normally derived form an empirical energy function, based on statistics derived for known protein structures. After the best-fit template is chosen, the structural model of the sequence is built based upon the alignment with the selected template. The protein threading method relies on the belief that the number of different protein folds in nature is limited, mostly as a result of evolution but also due to constraints imposed by the basic physics and chemistry of polypeptide chains. The protein threading methods can be broadly divided into two classes: first, methods that derive a one dimensional profile for each structure in the fold database and align the target sequence to these profiles and second, methods that consider the overall three dimensional structure of the protein template.

2.3.3 *Ab initio* folding

The tertiary structure of a target protein is built from scratch in an *ab initio* structure prediction method. In other words the structural model of a protein sequence is constructed from the primary sequence alone, without resorting to any parent structural template. The *ab initio* methods can be divided into two classes: one is the class of methods that predict the structural models by simulating the protein's folding pathway and the other are methods that directly predict the most probable structure of a protein by searching the entire conformation space of the sequence.

If we know the physical environment where the protein folds, then theoretically we can simulate the folding pathway by implementing the physical laws for atomic interactions [110, 159, 45]. Although there has been progress in the use of full-atom simulations with explicit and implicit solvent models to predict the folding of small protein sequences [61], the success rate of such methods are very limited because of two reasons. First, there is no complete and clear understanding of the underlying mechanisms of protein folding yet. Second, these approaches involve simulating the interactions of all the atoms in a system of many thousands of atoms for dozens of microseconds, at a time step of femtoseconds. These computational requirements are beyond the power of today's computers [45].

Unlike the simulation based approaches, the methods that search the conformation space directly have been more successful. The software package, Rosetta [147], is an outstanding example. However, the exhaustive search of the conformation space is still formidable

because of its huge size. Several techniques have been adopted to cope with this problem such as simplifying the representation of the proteins (minimalist models) or restraining the conformation space.

In all search based *ab initio* methods, some optimization technique must be used to minimize the energy function to find the most probable conformation. The commonly used techniques include Monte Carlo sampling, simulated annealing, and genetic algorithms [147, 79, 149]. Even though the *ab initio* prediction methods have been extensively researched in the past three decades [64], they are not as successful as the homology modeling and fold prediction methods due to their very low accuracies.

Chapter 3

Inverse protein folding

3.1 Introduction

The inverse protein folding (IPF) or computational protein design (CPD) arises in application such as drug design, nano-technology or other industrial applications such as designing enzymes. The inverse protein folding is the complement of the protein folding problem. The input to the IPF problem is a target structure (a protein-like structure) and the output is a protein sequence that satisfies the following three criteria [40, 166]. First the protein sequence should fold to the target conformation. This means that the native fold of the sequence should be the target conformation. This criteria is often called *positive design*. Second, the target conformation should be the only native fold of the sequence. This criteria is referred to as *negative design*. Third, there should be a large gap in the energy of the native (target) fold of the sequence and the energy of any other fold of the sequence. In some applications the second criteria is relaxed and sequences with a few native folds are also considered.

Protein design is important for two reasons [86]. First it gives an insight to the protein folding process. Although substantial experimental and theoretical progresses have been made in understanding the basic physical and chemical laws that define a protein structure, as well as towards unraveling the steps of the folding process itself, solving the protein folding problem remains the "holy grail" for computational structural biology. An alternative route that could lead indirectly to this goal is to study the inverted problem (IPF). Second novel proteins can be synthesized that exhibit novel activities. An example is the chemical addition of a toxin to antibodies specific for cancer cells so as to enable more efficient, targeted

treatment of tumors [128, 25, 89].

The folding problem and the inverse folding problem are related as the physical laws that govern folding also stipulate the protein sequence. However, protein design is different in that the inverse folding problem usually does not provide a unique sequence as an answer. The key question to be answered is: how many, and which sequences can fold into a given conformation? This involves a search in sequence space for sequences that make the native structure both stable and unique.

Historically, proteins have been designed by applying rules observed from natural proteins, or by employing selection and evolution experiments in which a particular function is used to separate the desired sequences from the pool of largely undesirable sequences. The first successes in protein design were based on manual inspection and heuristics gleaned from examining naturally occurring proteins [34, 33]. It was noticed early on that many of these designs differed from naturally occurring proteins in that they did not contain well-packed side-chains in their interior [12]. In fact, most of the present experimental approaches enjoyed only limited success, providing proteins that in most cases fold into compact but mostly disordered conformations of molten-globule-like species [137]. The limitations in experimental design seems to be the result of a relatively low synergism between experiment and theory [144].

Recently, computational methods have been proven to be effective in designing proteins with enhanced specificity and stability. These methods use algorithms to search combinations of side-chains and identify those that pack together most efficiently within the context of a given backbone conformation. These algorithms have been successful and have been used to stabilize proteins, solubilize membrane proteins, redesign protein-protein interactions, create new enzymes, and design novel protein structures [57, 47, 90, 107, 150]. The computational and experimental methods can be used in combination to produce successful designs. An important success story based on such synergism of theory and experiment is given in [31].

Computational design methods has been mostly used to redesign already existing protein structures. While this is an important problem and novel functional proteins have been created with this approach [47, 104], in the long run it will be advantageous to create proteins of novel structures. Designing novel protein structures is referred to as *de novo* protein design and is intrinsically more difficult than protein redesign because a priori it is not known if the target structure is designable.

Since the emergence of the protein design problem, several models have been introduced in the literature that focus on specific design aspects of the protein structures such as the design of hydrophobic cores [134, 69, 36, 63, 32], the design of the surface and inter-facial residues [32, 130], the design of metal Ion-Binding sites for functional proteins [70, 26] or the design of secondary and super secondary structures [142, 83]. The design methods specialized for subclasses of protein structures such as *globular* [54, 141, 158, 91] and *fibrous* [124] proteins have also been proposed.

Clearly, the overview of all of these models and methods is beyond the limits of this dissertation. Therefore, we will only review two major models: the SCR and the HP models.

3.2 Side-Chain Rotamer (SCR) model

The protein design problem can be formally stated as follows: Given a target conformation C, specified by the atomic coordinates of a backbone structure, find an amino acid sequence S that will fold to that structure. Since, there may be many structures that adopt the fold, to increase the chances of success, we will try to find one of the most stable sequences by minimizing the quantity $\Delta G^{fold} = G^{folded} - G^{unfold}$ where G^{folded} is the energy of the folded state of the protein and $G^{unfolded}$ is the energy of the unfolded state. Any attempt to solve this problem will face two major challenges. First there are astronomically many possible sequences and in general these cannot be enumerated exhaustively. Second to evaluate ΔG^{fold} for a sequence we need to know the energy in the folded and unfolded states. Even by knowing the backbone structure and the sequence of amino acids it is not easy to calculate the energy of the conformation as the side-chain of amino acids can assume different orientations which affects the energy of the fold. In fact determining the correct orientation of the side-chains given the coordinates of the target backbone and the sequence of the protein is a major sub-problem of protein structure prediction.

Therefore, in most computational protein design methods only a discrete set of amino acid states and side-chain orientations (called *rotamers*), relative to the target backbone structure, is optimized according to a scoring function [88]. The rotamers represent the statistically dominant orientations of amino acid side-chains in naturally occurring proteins [74, 135, 46]. We call this formulation of the problem the Side-Chain Rotamer (SCR) model.

Choosing one rotamer at each position defines the global conformation of all the atoms

in the system and implicitly specifies an amino acid sequence. Different conformations are ranked using an empirical potential function that attempts to quantify the free energy of the system [132]. Most of the time, for simplicity, the potential function is assumed to contain only pairwise terms, which may be used to describe van der Waals, electrostatic and hydrogen bonding interactions, as well as solvent exposure [58]. Therefore, the energy of a sequence folded into a target structure can be expressed as:

$$E_{folded} = E_{t'} + \Sigma_i E(i_r) + \Sigma_i \Sigma_{j,j<i} E(i_r, j_s) \tag{3.1}$$

where $E_{t'}$ is the template self energy (i.e., backbone energies or energies of rigid regions of the protein not subject to rotamer-based modeling), $E(i_r)$ is the rotamer/backbone energy for rotamer r of residue i and $E(i_r, j_s)$ is the rotamer/rotamer energy of rotamers r and s of residues i and j, respectively. However, by assuming that the energy between rotamers is pairwise as in equation 3.1, certain non-additive energy contributions cannot be treated exactly, such as a surface area-based solvation potential [153]. Finding a set of rotamers that minimizes equation 3.1 is called side-chain positioning (SCP) problem. Pierce and Winfree in [132] proved that SCP is NP-hard and Chazelle and Kingsford showed that it is even hard to approximate the SCP problem [21].

The computational protein design methods have been extensively reviewed in the literature [48, 141, 158, 35, 161, 75, 133, 52, 127, 12]. We will briefly review the major components of these methods in the following subsections.

3.2.1 Choosing the target conformation

The first step in protein design is defining the target conformation. This might be a naturally occurring protein fold (in case we need to redesign a natural protein), a novel fold, or a new protein-protein interaction. In lattice simulations, many different sequences fold into the same low-energy structure, whereas other possible folds are rarely if ever found to represent the lowest-energy structures for any sequence [144, 20]. This indicates that most randomly generated protein structures are not designable [12], therefore it is very important that the target structure poses many of the defining characteristics of naturally occurring proteins. Clearly if the target conformation is an existing protein fold this would not be an issue. However for novel structures much attention must be spent to make sure that it is designable.

If the target conformation is a naturally existing fold, the coordinates of its crystallographic or NMR structure are used. However, in de novo design, we need to create a

set of coordinates for a novel conformation. This may be done using a rigorous mathematical parametrization of target conformations that mostly contain small units of protein secondary structure such as, 222-symmetric 4-helix bundles [71, 51], β-sheets [140, 24], β-hairpins [94], TIM barrels [120, 23], or coiled coils [62]. Alternatively, a backbone structure may be assembled from libraries of folded structures [148].

3.2.2 Designing sequences for target backbones.

Once the backbone of the target conformation has been selected, the next step involves *inverse design* that is finding a sequence that will fold into the target structure. Several computational methods have been developed for solving this problem (for a review see [75]). All of these methods share two common components: (a) an energy function for evaluating the fitness of a particular sequence for a particular structure and (b) an algorithm for searching for low-energy sequences. The common energy functions and search protocols for protein design have been reviewed previously [58, 75, 113, 127, 133, 52].

3.2.3 Energy functions

Describing the interactions in a protein accurately is a key element to protein design. Energy functions for protein design must be fast and accurate, yet not oversensitive to the fixed backbone approximations and discreteness of the rotamer library [133]. The energy functions for CPD problem contain several terms that reflect the interactions of amino acid/amino acid and amino acid/solvent molecules. These interactions include van der Waals forces, hydrophobic interactions, and electrostatic interactions such as hydrogen bonding and salt bridges.

In general, protein design energy functions are constructed to favor close packing between amino acids, satisfy hydrogen-bonding potential, partition hydrophobic amino acids to the core of a protein and polar amino acids to the surface, and favor low-energy torsion angles [12]. Although accuracy of energy functions is an important requirement, fairly simple functions have worked well for designing protein cores. The reason is that cores are the easiest part of the protein to describe energetically if one restricts the composition to hydrophobic residues. Packing is often evaluated with a Lennard-Jones (LJ) potential. Lennard-Jones potential has a mild attractive term and a strong repulsive term. The attractive portion of the LJ potential models van der Waals forces and draws atoms near each

other. The repulsive portion of the potential ensures that the atoms do not become too close.

Despite the van der Waals energies that are described quite well by molecular mechanics force fields, electrostatics, solvation, and hydrophobic/polar interactions are hard to model by molecular mechanics force fields [48]. One reason is that, water itself is an extremely complicated substance to model and parametrize because of its polarizability, interactions with polar groups, and entropic contributions due to the hydrophobic effect [133]. The hydrophobic effect is usually modeled by assuming that the penalty for exposing non-polar groups to water is dependent on the surface area [50] and the solvation energy is calculated from the change in solvent accessible surface area, multiplied by an atom-dependent atomic solvation parameter. These solvation parameters are typically derived from transfer free energies of amino acids between water and vacuum or some organic solvent.

Hydrogen bonding interactions play an important role in stabilizing secondary structures and imparting specificity to proteins. Hydrogen bonds are taken into account with an explicit hydrogen bonding term in some force fields, such as DREIDING or they are accounted for through the electrostatics and van der Waals energies in other force fields such as AMBER, OPLS, and CHARMm [133].

Other terms such as secondary structure propensities have also been incorporated in energy models for sequence design [29, 117, 116].

3.2.4 Searching methods

Although discretizing the side-chain rotation configurations makes the search space discrete and computationally feasible the sequence/structure space is still large. For a 100-residue protein in which all 20 amino acids are permitted at every position, with only two rotatable bonds and five conformations each, there are 500^{100} sequence/structure solutions [133]. Clearly, exhaustive search of the solutions space is not an option even for moderate size protein sequences. Therefore, several methods have been developed to selectively search the sequence/structure space for finding near optimal solutions. The strengths and weaknesses of various search algorithms is reviewed in [35], and implementations of these algorithms are evaluated quantitatively in [161].

These methods can be divided into two categories: *Stochastic* and *Deterministic* methods. *Stochastic* methods such as Monte Carlo (MC) and genetic algorithms (GA) semi-randomly sample solutions and then move from one possible solution to another in a manner

that depends on both the nature of the energy landscape and the algorithm-specific rules for movement. While these algorithms can be applied to the design of long protein sequences that have virtually an infinite number of possible solutions, there is no guarantee that they will explore solutions near the global energy minimum.

In contrast, methods that fall into the second category, the deterministic methods, are intended to be functionally equivalent to an exhaustive search therefore, they ensure that the global minimum energy configuration (GMEC) is identified when they converge. However, since truly exhaustive searches are possible only for very small search spaces, deterministic algorithms prune the search space by applying rejection criteria in order to eliminate the vast majority of combinatorial possibilities without actually considering them formally. Clearly, the robustness of these methods depends both on how finely the conformational space is represented and on the criteria used for rejection but they are mostly slow and are not suitable for designing long proteins.

3.2.5 Stochastic methods

Stochastic methods that are applied in protein design explore the solution space by altering side-chain identity, side-chain orientation and backbone structure. The simplest type of stochastic methods is the Monte Carlo (MC) method. The general strategy of MC algorithms is to iteratively propose a modification to the current solution and then decide whether or not the proposed modification should be accepted. The most common way of deciding whether to accept a proposed modification is to use the Metropolis criterion [114]. The initial solution is constructed by randomly choosing the rotamers for a sequence. Then, a rotamer substitution is made at a randomly picked residue in the sequence. Rotamers of different amino acids are treated equally, so a rotamer substitution can be either the same amino acid or a new one. A new energy E_{new} is calculated and if this energy is lower than the previous energy E_{old}, the modification is accepted. If the energy is higher, the modification is accepted with the Boltzman probability

$$p = e^{-\beta(E_{new}-E_{old})}, \beta = \frac{1}{kT} \tag{3.2}$$

where k is Boltzman's constant. The role of the temperature T is to avoid being trapped in multiple local minima in the energy landscape by allowing the trajectory to surmount energy barriers. To strengthen this effect, an initial temperature is selected and annealed. The temperature is then cyclically raised and lowered over the course of a single run between

a designated high and low temperature. The MC methods have been used extensively in protein design because of their simplicity and satisfactory performance on difficult energy landscapes [72, 69, 30, 55, 91].

GA methods are similar to MC methods in the sense that they also propose modifications to the intermediate solutions and mostly accept those which result in better solutions. The major distinction is that instead of modifying one intermediate solution in each step, a population of solutions is evolved throughout the genetic operators, such as recombination, that create new solutions from existing ones. GA methods are reported to be efficient due to the implicit parallelism contained within protein design problems; different segments of the structure are optimized in parallel and selective recombination between models will sometimes bring two of the optimized segments together into the same model [35]. GA methods have been applied to a wide range of problems, including protein structure prediction [129] and protein design [78, 36, 98]. The advantages of GA methods are that the population dynamics can handle local minima problem more efficiently by making moves in solutions space that are larger than the moves typically made by MC methods. In addition, beneficial mutations can be combined onto a single solution, increasing the number of paths that circumvent local minima. As a disadvantage, GA methods do not perform well on highly coupled systems where crossover disruption is problematic, as is expected for side-chain systems. Furthermore, residues that are close in sequence are not necessarily close structurally, making it difficult for the algorithm to find clean crossover points [161]. Both the MC and GA are relatively straightforward to be incorporated in protein design algorithms however, they require an intensive optimization of the parameters to control the convergence properties of the algorithm, with respect to the system being studied.

Other stochastic methods have been developed to be used in protein design among which FASTER [39, 3] is the most important. While the deterministic methods are overly cautious in the elimination of rotamers and other stochastic methods may be too crude for the combinatorial nature of the problem, FASTER is designed to perform in the middle spectrum.

3.2.6 Deterministic methods

The most commonly used pruning idea currently used in the design of protein sequences is based on the application of the dead end elimination (DEE) theorem [37]. In simple terms, the DEE theorem allows individual side-chain rotamers to be strictly designated

as being incompatible with the global energy minimum. DEE is fundamentally based on the following physical concept. Consider two rotamers, i_r and i_t, at residue i and the set of all other rotamer configurations S at all residues excluding i of which rotamer j_s is a member. If the pairwise energy contributed between i_r and j_s is higher than the pairwise energy between i_t and j_s for $s \in S$ for all S, then i_r cannot exist in the GMEC and can be eliminated. This notion is expressed mathematically by the inequality:

$$\forall s \in S \;\; E(i_r) + \Sigma_{j \neq i} E(i_r, j_s) > E(i_t) + \Sigma_{j \neq i} E(i_t, j_s) \tag{3.3}$$

If the above condition holds, the rotamer r at residue i can be ignored because it cannot be part of the GMEC. The inequality 3.3 is not computationally tractable because, to make an elimination, it is required that the entire sequence/rotamer space be enumerated. To simplify the problem we can reformulate it as follows:

$$E(i_r) + \Sigma_{j \neq i} \min_s E(i_r, j_s) > E(i_t) + \Sigma_{j \neq i} \max_s E(i_t, j_s) \forall \{S\} \tag{3.4}$$

Using an analogous argument, equation 3.4 can be extended to the elimination of pairs of rotamers inconsistent with the GMEC [161] as follows.

$$\epsilon(i_r, j_s) + \Sigma_{k \neq i,j} \min_t \epsilon(i_r, j_s, k_t) > \epsilon(i_u, j_v) + \Sigma_{k \neq i,j} \max_t \epsilon(i_u, j_v, k_t) \tag{3.5}$$

where ϵ is the combined energies for rotamer pairs:

$$\epsilon(i_r, j_s) = E(i_r) + E(j_s) + E(i_r, j_s) \tag{3.6}$$

and

$$\epsilon(i_r, j_s, k_t) = E(i_r, k_t) + E(j_s, k_t) \tag{3.7}$$

The singles and doubles DEE criteria in their original form may fail to discover special conditions that lead to the determination of more dead ending rotamers. For instance, if the energy contribution of rotamer i_t is always lower than i_r without the maximum of it being below the minimum of i_r the criteria imposed by inequality 3.4 cannot capture the dead ending of rotamer r at residue i. To address this problem, Goldstein [56] presented a modification of the criteria as follows:

$$E(i_r) - E(i_t) + \Sigma_{j \neq i} \min_s \{E(i_r, j_s) - E(i_t, j_s)\} > 0 \tag{3.8}$$

The calculation time of the double eliminations is significantly more than the single eliminations. To accelerate the process, efficient methods have been developed to predict

the doubles calculations that will be the most productive [59]. These modifications, referred to as fast doubles, significantly improved the speed and effectiveness of DEE.

Several additional modifications collectively enhance DEE further. For instance in [38, 56] rotamers from multiple residues are combined into so-called super-rotamers to achieve further eliminations. This way multiple rotamers can be eliminated in a single step. In addition, it was shown that "splitting" the conformational space between rotamers improves the efficiency of DEE [131].

The DEE based methods find the GMEC if they converge, however in most applications it is very hard, if not impossible, to devise efficient DEE criteria that eliminate all non optimal sequence/rotamer conformations. As a post-processing step the conformations that survive DEE will be energy-minimized, in such cases. When energy minimization is performed after pruning with DEE, the combined protein design process becomes heuristic, and is no longer provably accurate. In [53] the dead-end criterion was extended to apply to continuous deformation of rotamers through redefining each rotamer as representing a continuous voxel in a local conformational space.

Self-consistent mean field (SCMF) method is another deterministic algorithm used to search the sequence/rotamer conformation space in protein design problem. SCMF uses a mean-field description of the rotamer interactions to alter the energy landscape which in turn is used to determine the probabilities of each rotamer in each position. The method iteratively develops a probabilistic description of the relative population of each possible rotamer at each position. The probability of a given structure is defined as a function of the probabilities of its individual rotamer components. The most probable structure is returned as the GMEC when the updating cycle is terminated. Although SCMF uses a probabilistic description to determine the optimal configuration, it is in fact a deterministic method as it always converges to the same solution given a set of run parameters.

The mean-field energy for rotamer i_r at residue i is defined as follows [85]:

$$E_{mf}(i_r) = E(i_r) + \Sigma_{j \neq i}^{N} \Sigma_{s=1}^{K_j} E(i_r, j_s) V(j_s) \tag{3.9}$$

where K_j is the total number of rotamers at residue j, $V(j_s)$ is the conformational probability vector which is normalized to unity. The first term in equation 3.9 is the contribution due to the interaction between the rotamer and the backbone, and the second term describes all the inter-rotamer pairwise interactions weighted by the probability of that rotamer existing in the GMEC. The conformational probability vector can be independently calculated by

Gibb's ensemble as follows [161]:

$$V(j_s) = \frac{1}{q_j} e^{\beta E_{mf}(j_s)} \tag{3.10}$$

where q_i is the partition function:

$$q_j = \Sigma_{s=1}^{K_j} e^{\beta E_{mf}(j_s)} \tag{3.11}$$

This equation smooths the landscape and avoid the problem of multiple local minima, making it relatively simple to locate the minimum of the mean-field energy landscape. The mean-field energy is minimized using an annealing method explained in [99]. The initial temperature for the annealing process is set to a high value (ofter more than $20,000K$) and the probability vector $V(j_s)$ is set to $1/K_j$, thereby assigning equal probability to each rotamer. When the initialization is done, the mean-field potential $E_{mf}(i_r)$ is calculated from equation 3.9 for each residue and rotamer. These energies are used to calculate the probabilities using Gibb's equations. The algorithm iterates between equations 3.9 and 3.10 until the energy converges and self-consistency is achieved. A convergence criterion of 0.0001 for $V(j_s)$ is used in [85] to define self-consistency. The temperature is then lowered in linear increments of $100K$ and the procedure repeated. When the final temperature is reached ($100K$), the conformational vector represents the probability of each rotamer at a given residue position. The best solution is determined as the collection of rotamers that have the highest probability at each position [161].

Unfortunately, there is no guarantee that the minimum of the mean-field landscape corresponds with the true GMEC. However, the advantage of SCMF is that the computational time scales linearly with the number of residues, making it possible to obtain solutions for proteins unattainable by DEE based methods.

3.3 Inverse Protein Folding in HP models.

A critical step in the folding pathway of globular proteins is the formation of a tightly packed hydrophobic core. This core is formed due to hydrophobic interactions that draw the hydrophobic (water repelling) amino acids together and drive the hydrophilic (water attracting) amino acids to the surface of the protein. It is believed that hydrophobic interactions are the major driving forces in determining the native tertiary conformation of proteins. Based on this, Dill [41] introduced the simplified model of hydrophobic-polar

(HP) for protein prediction and design problems. Two major simplifying assumptions are made in the HP model: first the 20 amino acids from which proteins are made are replaced by two types of monomers: hydrophobic (H, "1") and polar (P, "0"). Second the protein sequence is laid out on a spatial lattice with each monomer occupying exactly one vertex of the lattice and neighboring monomers occupy neighboring vertices such that the chain of monomers makes a self avoiding walk in the lattice.

Although the HP models are very simplistic they are known to adequately describe proteins at the coarse-grained level with the advantage that the native states can be determined exactly [97, 18, 42, 125, 11, 14]. Furthermore, they provide a controlled setting for theoretical analysis and rigorous testing of concepts and ideas for future use in studies on real proteins [115, 167]. The importance of the minimalist models and their applications in forward and inverse protein problems has been extensively reviewed in [87].

In the standard HP model [96, 97, 18] the energy of a sequence S in a lattice conformation G is simply given by the negative of the number of contacts between adjacent pairs of H monomers which are not consecutive in the chain. This contact energy function can be formally defined as follows: let a target conformation be described by a graph $G = (V, E)$ with vertices V, that correspond to amino acids, and edges $E \subset V \times V$, that define a self-avoiding walk on a 2D or 3D lattice. We construct a contact graph $\bar{G} = (V, \bar{E})$ induced by G, where an edge (u, v) is in $\bar{E}$ if $u, v \in V$, $(u, v) \notin E$ and (u, v) is an edge in the lattice. We call $\bar{E}$ the contacts set. For a sequence S, let $E(G, S)$ be the conformational energy of S when the vertices of G are labeled with the sequence of amino acids defined by S. Then $E(G, S) = -|\hat{E}|$ where $\hat{E} = \{(u, v) \in \bar{E} |\ u$ and v are labeled with $H\ \}$. The edges in $\hat{E}$ are called *HH contacts*.

Given an input protein sequence S the goal of protein prediction problem is to find a conformation G^* that produces the minimum value for $E(G, S)$ in the HP model settings. This problem is known to be NP-hard in 2D square [27] and 3D cubic lattices [7].

The input to the inverse protein problem is a target conformation G and the goal is to find a sequence S^* that first has the minimum energy in G (positive design) and second, the sequence does not fold to (too many) other conformations (negative design). The IPF can be defined more precisely as follows [66]. Let S_G be the set of sequences that achieve their lowest energy in the target conformation G. Formally,

$$S_G = \{S \in \{H, P\}^n \mid E(G, S) \leq E(G', S), \forall G' \in \mathcal{G}_n\} \tag{3.12}$$

where $n = |V|$ and $\mathcal{G}_n$ be the set of all target conformations for which the length of the chain is n. We say that a sequence S folds to G if $S \in S_G$. The degeneracy of a sequence refers to the number of conformations for which the sequence assumes its lowest energy. Formally,

$$d(S) = |\{G \in \mathcal{G}_n \mid E(G,S) \leq E(G',S), \forall G' \in \mathcal{G}_n\}| \tag{3.13}$$

The sequences with degeneracy of one are called *stable* proteins. Let $\delta_G = \min_{S \in S_G} d(S)$; this is the minimal degeneracy possible for a sequence S that folds to G. Now given a conformation $G = (V, E)$ embedded in a 2D or 3D lattice, the goal in IPF is to find a protein sequence $S \in \{H, P\}^n, n = |V|$, such that $S \in S_G$ and $d(S) = \delta_G$.

Note that although proteins are generally assumed to have a unique lowest energy conformation (be stable), this formulation of IPF explicitly recognizes that for a given target conformation it may not be possible to find a stable sequence that folds to the target conformation. For instance the degeneracy of any sequence that folds into a conformation with empty set of contacts is 2^n where n is the size of the conformation. For conformations with non-stable solutions, the most reasonable goal is to find a sequence with minimal degeneracy.

Unlike the protein prediction problem, the complexity of the IPF as defined above is not known but it is conjectured that the problem is NP-hard [8]. Several heuristic based algorithms have been introduced for solving the IPF problem in HP model [40, 93, 155, 82, 146, 166]. Although these methods do not guarantee to produce the exact solution, they attempt to capture the positive and negative design aspects of the IPF.

The heuristic methods can be divided into two categories: first, those that use observations about the properties of proteins to justify algorithms that design sequences [82, 166]. The second category of heuristic methods are those in which an alternative formulation of IPF is proposed [40, 93, 146, 155]. These alternative formulations attempt to capture the positive and negative design issues by defining a heuristic sequence design (HSD) problem. An implicit assumption of this approach is that a sequence that satisfies the HSD problem is likely to solve IPF. Ideally the proposed HSD problems should be solvable efficiently, however as we will see later this is not always the case.

We will review two major HSD problems the *canonical* [146] and the *grand canonical* [155] models.

3.3.1 Canonical model

The canonical model was introduced by Shahknovich and Gutin in [146]. They observe that for any target conformation the conformational energy can be minimized simply by choosing hydrophobic monomers for all positions. However, this sequence is unlikely to achieve its lowest energy with the target conformation. Therefore, to deal with this they limit the number of hydrophobic monomers that can be used in a protein sequence. Hart formulated this HSD problem in [66] as follows: Given a target conformation $G(V, E)$ in a 2D or 3D lattice L, the protein design in canonical model is the problem of minimizing $E(G, S)$ subject to the constraint that no more than $[\lambda n]$ hydrophobic monomers be used to design S, where λ is a real value less than 1. Shahknovich and Gutin used [146] a stochastic Monte Carlo based method to solve the protein design in 3D cubic lattice under the canonical model. The complexity of IPF under canonical model was investigated in [66] and it was shown that the problem is NP-hard in both 2D and 3D cubic lattices. However, Berman and et al. [8] showed that the reduction that was used in [66] to prove the NP-completeness of the problem on 2D lattice was incorrect. They devise an efficient algorithm for solving the problem on 2D square lattice. They also showed that the canonical IPF is indeed NP-hard in 3D cubic lattice however, a polynomial time approximation scheme was designed for the problem.

The canonical IPF problem is in fact a special case of the *Densest Subgraph* (DS) problem defined as follows. Given a graph $G = (V, E)$ and a positive integer K as inputs, find a $V' \subseteq V$ with $|V'| \leq K$ that maximizes $|\{(u, v) \in E : u, v \in V'\}|$. Clearly, the canonical IPF problem can be reduced to DS problem by setting $K = [\lambda n]$ and G to the contact graph of the target conformation. The DS problem in its general form is NP-hard however, it can be solved in $O(K|V(G)|)$ when the input graph is the contact graph of a target conformation in 2D square lattice [8].

The canonical IPF in 3D cubic lattice was shown to be NP-complete by a reduction from the maximum clique problem to the DS problem on general graphs [8]. The challenging part is to make sure that the reduction works for the special topology of the input contact graph and that such a contact graph can in fact be realized by a 3D sequence.

3.3.2 Grand Canonical model

In the grand canonical (GC) model, introduced in [155], the number of hydrophobic monomers is limited by adjusting the energy function instead. For instance, in [66] a lattice based formulation of the GC model was considered in which every hydrophobic contact contributes -2 to the total energy, every solvent accessible site of a hydrophobic amino acid contributes 1, and all other interactions do not contribute to the total energy. Since the hydrophobic monomers are penalized for their exposure to solvent, this contact potential implicitly limits the number of hydrophobic monomers in the sequence. Hart [66] gave an optimal $O(n^2)$ algorithm for solving the IPF in GC model based on this formulation.

In the generalized formulation of the GC model the energy function is defined by the following equation [84]:

$$E(G,S) = \alpha\Sigma_{i,j\in S_H, i<j-2} g(d_{ij}) + \beta\Sigma_{i\in S_H} s_i \tag{3.14}$$

where S_H denotes the set of positions in S that contain H monomers , $\alpha < 0$ and $\beta > 0$ are scaling parameters, s_i is the area of the solvent-accessible contact surface for residue i, d_{ij} is the distance between the residues i and j (in Å) and

$$g = \begin{cases} 1/[1 + exp(d_{ij} - 6.5)] & \text{when} \quad d_{ij} \leq 6.5 \\ 0 & \text{when} \quad d_{ij} > 6.5 \end{cases} \tag{3.15}$$

is a *sigmoidal* function.

The scaling parameters α and β have default values -2 and $\frac{1}{3}$, respectively. Notice that the conformations in this formulation are not limited to the vertices of a lattice. Kleinberg in [84] gave an optimal solution for this problem that runs in $O(n^2 \log n)$. Furthermore, he investigated an extension to the GC model by allowing fractional hydrophobicity for the amino acids with the hope that it provides an interesting contrast to the discrete HP model. In this definition each residue at position i is allowed to specify a hydrophobicity value z_i, where z_i is an arbitrary real number in the interval $[0, 1]$. Thus, a protein sequence in this model would be a sequence S of n real numbers, each between 0 and 1. The penalty for exposing residue i to solvent could be scaled by the hydrophobicity z_i, and the reward for a pairwise hydrophobic contact between i and j could be scaled by a product of the form $z_i z_j$. Taking these into account he redefined the energy function as follows:

$$E'(G,S) = \alpha\Sigma_{i<j-2} z_i z_j g(d_{ij}) + \beta\Sigma_{i\in S_H} s_i \tag{3.16}$$

Note the standard GC model is precisely the case in which each z_i is required to be either 0 or 1. Surprisingly, it turned out that this extension will not add any contrast to the discrete HP model as it was shown that for any target structure, with associated fitness function $E'(G, S)$, there exists an optimal sequence S in which each z_i takes the value 0 or 1.

When this extension failed, Kleinberg [84] considered a different extension in which a finite alphabet of amino acids $\{a_0, a_1, \ldots, a_k\}$ is considered, where a_0 is designated as the most polar residue type and a_k the most hydrophobic residue type. The energy function in this model is define as:

$$E''(G, S) = \alpha \Sigma_{i<j-2} \epsilon_{t_i t_j} g(d_{ij}) + \beta \Sigma_{i \in S_H} \delta_{t_i} s_i \tag{3.17}$$

where t_i is the residue at position i, $\epsilon_{t_i t_j}$ is the contact parameter that indicates the reward for having a contact between residues of type t_i and t_j and δ_{t_i} is the solvation parameter that indicates the penalty for exposing residue of type t_i to solvent. The IPF problem is proved to be NP-hard for general set of parameters under this formulation however, Kleinberg showed that it is possible to design optimal sequences efficiently with respect to a large class of parameter sets [84].

Note that the definition of canonical and grand canonical model do not grantee either positive or negative design criteria.

Chapter 4

Structure approximating IPF in 2D square lattice

4.1 Introduction

Although the canonical and grand canonical models are defined to capture the positive and negative aspects of protein design, none of them actually guarantee that the designed sequences satisfy the negative and positive criteria. In Gupta *et al.* [60], the IPF problem was studied from a different perspective. Instead of designing a sequence directly for the target fold and relaxing conditions on the sequence, they introduced a design method in 2D square lattice under the HP model that can approximate any target conformation and showed that approximated structures, called *constructible* structures, are native for designed proteins (positive design). However the main challenge that is to prove the stability of the designed proteins (negative design) remained largely unsolved. They conjectured that the constructible structures are stable and proved it for two very simple but infinite subclasses of constructible structures, namely for L_0 ("I"-shape) and L_1 ("L"-shape) structures (cf. Figure 4.4).

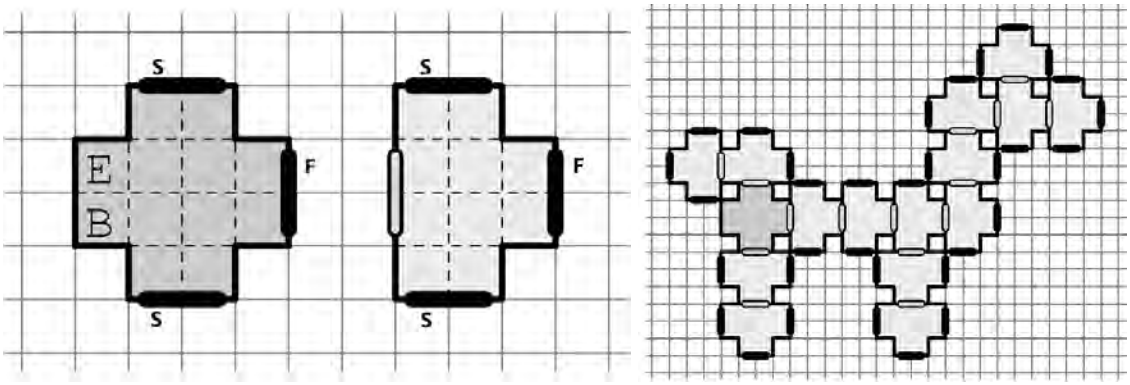
Our goal is to solve the conjecture for a subclass of constructible structures that is rich enough to approximate (although more coarsely) any given shape. The major difficulty of this task is the analysis of a large number of cases that arises in the proof process. As an evidence the stability proof of L_0 and L_1 structures presented in [60] requires 2 and 6 pages of case analysis, respectively, although they are very simple subclasses of constructible

structures. We attempt to overcome this problem first by incorporating additional effective forces in the folding process into the HP model and second by developing a program (2DHP-Solver) for semi-automatic analysis of the cases that arise in the proof process. Along these lines, we extend the HP model by adding a third type of monomers, the cysteines, in the designed protein sequences and incorporating disulfide bridges between cysteine monomers in the energy function. We call this model the Hydrophobic-Polar-Cysteine (HPC) model.

In the following sections we show that these two strategies are enough to prove the stability of a rich subclass of constructible structures namely the *wave* structures. As a first step we introduce a rich subclass of constructible structures called *snake* structures and prove that the proteins of snake structures are stable under the *strong* HPC model, an artificial version of the HPC model. We then prove the stability of the wave structures a subclass of snake structures under the proper HPC model.

4.2 Constructible structures

The constructible structures are formed by a sequence of tiles. There are two types of tiles: a starting tile in the shape of "+", and a regular tile in the shape of "⊢", depicted in Figure 4.1(a). Both tiles have three *ligands* depicted with black lines, two of which are side ligands marked with "S" and one forward ligand marked with "F". In addition, the regular tile has one *receptor*, depicted with a gray line.



(a) (b)

Figure 4.1: (a) The starting tile (left) and the regular tile (right) for constructible structures (b) An example of constructible structure: a tree built from basic tiles.

A constructible structure is a partial tiling of the 2D square lattice $\mathcal{L}$ obtained by the

following procedure:

1. Place the starting tile into the square lattice.
2. Place a regular tile into the square lattice so that its receptor is attached to a ligand of a tile already in the square lattice and it does not overlap with any other tile.
3. Continue with step 2., or end the procedure.

An example of a constructible structure is shown in Figure 4.1(b). A constructible structure is called *linear* if it is constructed such that every regular tile is attached to the ligand of the last placed tile. Therefore a linear structure can be represented as as sequence of tiles $< t_1, t_2, ..., t_n >$, where t_i is attached to two tiles t_{i-1} and t_{i+1} for $i \in [2, n-1]$ while t_1 and t_n are attached to the tiles t_2 and t_{n-1}, respectively.

Figure 4.2 depicts two linear constructible structures. In a linear constructible structure a tile t_i for $i \in [1, n-1]$ is called a *bending* tile if it is attached to t_{i+1} thorough one of its side ligands. We can classify the linear structures by the number of bending tiles they contain such that L_i represents the class of linear structures with exactly i bending tiles.

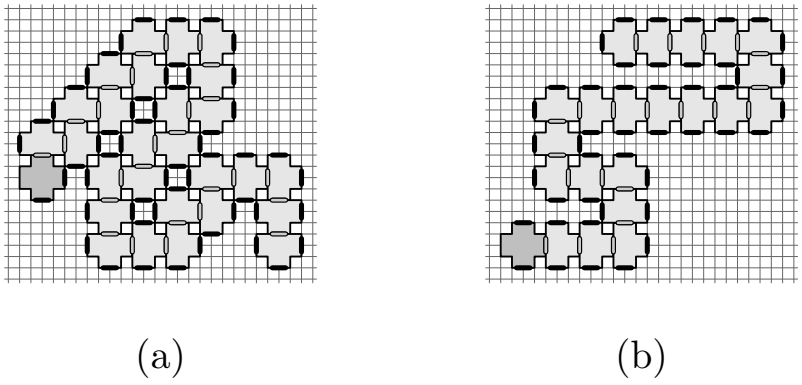
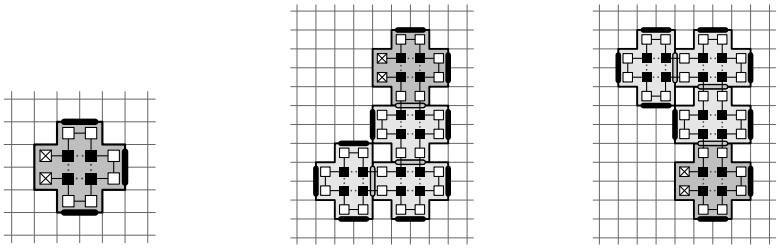


Figure 4.2: Two linear constructible structures.

After a constructible structure is formed to cover a target shape, a protein sequence is assigned to it by filling the constructing tiles with appropriate HP subsequences as follows. To each regular tile we assign an HP subsequence containing 4 hydrophobics surrounded by 6 polar monomers and to a starting tile we assign a subsequence with 4 hydrophobics surrounded by 8 polar monomers (cf. Figure 4.3). The resulting protein sequence has a native fold that exactly fills the corresponding constructible structure.

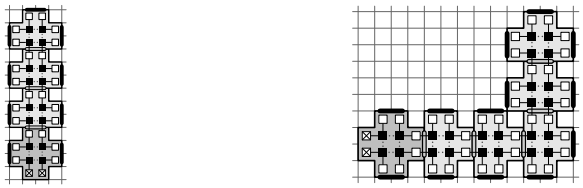


(a) (b)

Figure 4.3: (a) Assigning sequences to tiles. (b) An example of constructible structure with assigned protein sequence.

The proteins of constructible structures have a special property that makes it much easier to prove their stability. Every hydrophobic monomer in a protein of a constructible structure has two HH contacts which is the maximum number of contacts a hydrophobic monomer can make in any fold of the protein. Therefore, the constructible structures has the minimum possible energy with respect to the number of hydrophobic monomers. Such conformations are called *saturated* conformations. Clearly, the saturated conformations are native (positive design) and any native conformation must be saturated.

Gupta *et al.* [60] conjectured that the constructible structures are stable. The proof of the stability of any constructible structure is extremely hard. Thus they only proved that the proteins of two very simple subclasses of linear structures, namely L_0 and L_1 structures (cf. Figure 4.4) are stable. The stability of over 48,000 structures (including all structures with up to 9 tiles) was computationally tested as well. The stability proof of L_0 and L_1 structures is based on the induction on the rectangular boundaries that enclose all the hydrophobic amino acids of the saturated conformations of an L_0 or L_1 structure. Furthermore, they conjectured that it might be easier to prove the stability of the linear constructible structures.



(a) (b)

Figure 4.4: (a) An example of "I"-shaped constructible structure. (b) An example of "L"-shaped constructible structure.

Design of stable proteins of arbitrary lengths in the HP model was also studied by Aichholzer *et al.* [1] (for 2D square lattice) and by Li *et al.* [101] (for 2D triangular lattice), motivated by a popular paper of Brian Hayes [68].

4.3 Hydrophobic-polar-cysteine (HPC) model

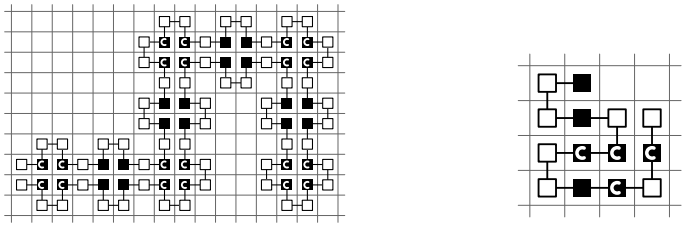
Cysteine is a hydrophobic amino acid [121] which contains a *thiol* group that can bind with the thiol group of another cysteine and form a *disulfide* bond or bridge. Disulfide bridges are the other significant forces in the folding process of the proteins which play an important role in the stability of the protein structure [73, 81]. We extended the HP model by adding a third type of monomers, cysteines (C or '2'), to the designed protein sequences and incorporating disulfide bridges between two cysteines into the energy model. We call this model the hydrophobic-polar-cysteine (HPC) model. We represent a protein chain in HPC model as a sequence $p = p_1 p_2 \dots p_{|p|}$ in $\{0, 1, 2\}^*$, where "0" represents a polar monomer, "1" a hydrophobic non-cysteine monomer and "2" a cysteine monomer.

In the HP model only hydrophobic interactions between two adjacent hydrophobic monomers which are not consecutive in the protein sequence (*hydrophobic contacts*) are considered in the energy model, with each contact contributing -1 to the total energy. In the HPC model cysteines act as hydrophobic monomers and contribute in the hydrophobic contacts. In addition, two adjacent non-consecutive cysteines can form a disulfide bridge contributing with another -1 to the total energy. However, each cysteine can be involved in at most one bridge. More formally, any two adjacent non-consecutive cysteines form a *potential* disulfide bridge and the disulfide-bridge energy is equal to -1 times the number of pairs in the maximum matching in the graph of potential disulfide bridges. The total energy of the fold is calculated as -1 times the number of contacts plus -1 times the number of bridges. For example, the energy of the fold in Figure 4.5(b) is $-5 - 1 = -6$. Our results show that adding cysteine in the protein sequences can indeed help in stabilizing the designed structures (see Chapter 5).

4.4 Snake structures

In this section we introduce a rich subclass of linear constructible structures called the *snake* structures. The snake structures are linear constructible structures in which every odd tile

is a bending tile. The hydrophobic monomers of the bending tiles and the terminal tiles are set to be cysteines, and all other hydrophobic monomers are non-cysteines, cf. Figure 4.5(a).



(a) (b)

Figure 4.5: (a) An example of a snake structure. The bending tiles use cysteines (black squares marked with C). (b) An example of energy calculation of a fold in the HPC model. There are 5 contacts between hydrophobic monomers, thus the contact energy is -5. There are three potential sulfide bridges sharing a common vertex, hence only one can be formed. Thus the sulfide bridge energy is -1 and the total energy is $-5 - 1 = -6$. The energy of this fold under the strong HPC model is -7 as it has one non-cysteine bridge.

Note that the snake structures can still approximate any given shape, although more coarsely than the linear structures. The idea of approximating a given shape with a linear structure is to draw a non-intersecting curve consisting of horizontal and vertical line segments. Each line segment is a linear chain of basic tiles depicted in Figure 4.1(a). At first glance, the snake structures seem more restricted than linear structures, as the line segments they use are very short and have the same size (3 tiles long). However, one can simulate arbitrary long line segments with snake structures forming a zig-zag pattern, cf. Figure 4.6.

4.4.1 The strong HPC model

We conjecture that the proteins of snake structures are stable in the HPC model, and furthermore it can be proved with techniques that we will present in the following sections.

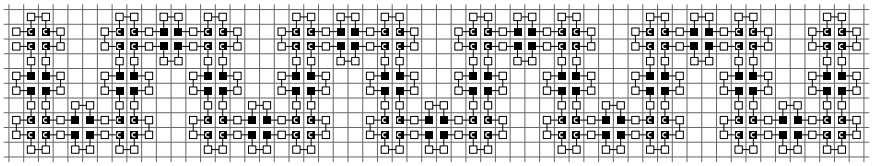


Figure 4.6: Simulation of a straight line segment with a snake structure.

As an evidence to the correctness of this conjecture, we present the proof that the snake proteins are stable in an artificial variant of the HPC model called *strong* HPC model. In this model, the energy function consists of three parts (first two are the same as in the HPC model): (i) the contact energy, (ii) the SS bridge energy and (iii) non-cysteine bridge energy. The last part is equal to -1 times the number of pairs in the maximum matching of the graph of potential non-cysteine bridges. There is a potential non-cysteine bridge between any two adjacent ordinary hydrophobic monomers. Thus, the fold in Figure 4.5(b) has energy $-5-1-1=-7$ in the strong HPC model. This energy model can be interpreted as follows: we assume that we have two types of cysteine-like hydrophobic monomers each forming bridges, but no bridges are possible between cysteines and hydrophobic non-cysteine monomers.

In the snake structures, approximately 40% of all monomers are hydrophobic and half of those are cysteines. Thus approximately 20% of all monomers are cysteines. Although, most of the naturally occurring proteins have much smaller frequency of cysteines, there are some with the same or even higher ratios: 1EZG (antifreeze protein from the beetle [102]) with 19.5% ratio of cysteines and the protein isolated from the chorion of the domesticated silkmoth [139] with 30% ratio.

4.5 Wave structures

Despite the fact that the snake structures are more restricted, the proof of their stability under the strong HPC model still required the analysis of a huge number of cases and this number rapidly increases in the proper HPC model. Therefore, we consider a subclass of the snake structures, called the *wave structures* and formally prove that they are stable under the proper HPC model.

The wave structures are instances of the snake structures that do not contain an occurrence of the four forbidden motifs in Figure 4.7. The wave structures can be constructed using a set of four *super-tiles* and their flipped versions (cf. Figure 4.8). Each super-tile has at most one position called "receptor", which connects to the next super-tile and at most two positions called "ligands", which can connect to the previous super-tile. The *starting* super-tile has one receptor and consists of two basic tiles (Figure 4.8(a)), the *terminating* super-tile has one ligand and consists of 5 basic tiles (Figure 4.8(b)), the *bending* super-tile has one ligand and one receptor and consists of two tiles (Figure 4.8(c)), and finally the

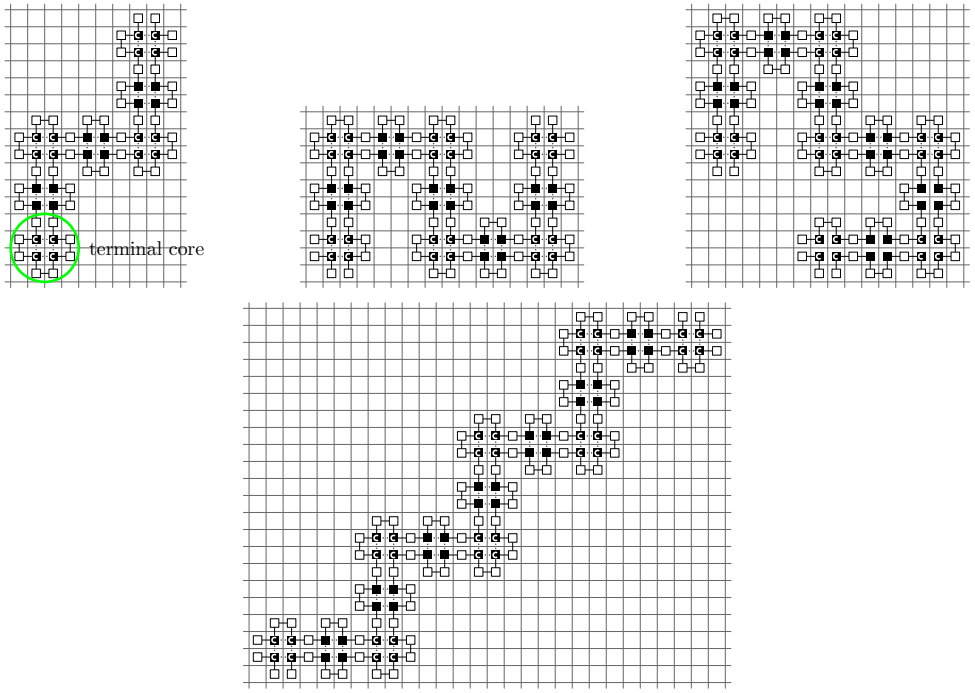


Figure 4.7: Forbidden motifs in wave structures.

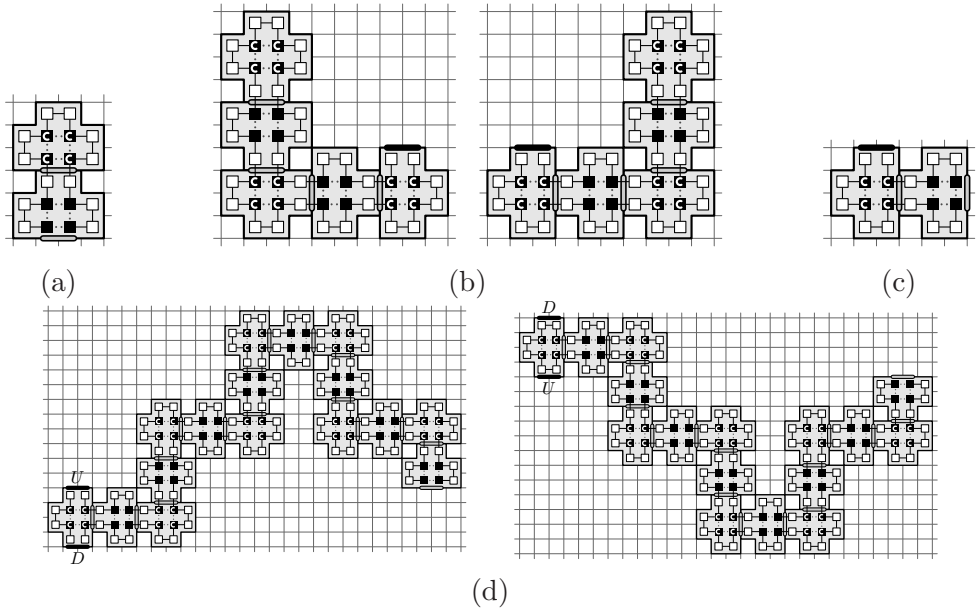


Figure 4.8: Super-tiles used to construct wave structures: (a) starting super-tile; (b) non-flipped and flipped versions of terminating super-tile; (c) bending super-tile; and (d) flipped and non-flipped versions of regular tile. The receptors are depicted with white ovals and ligands with black ovals.

regular super-tile has two ligands ("U" and "D") and one receptor and consists of 16 basic tiles (Figure 4.8(d)). The receptor of one super-tile can connect to the ligand of another one however, a regular super-tile must only connect through one of its ligands. A wave structure is a partial tiling of the 2D square lattice obtained by the following procedure.

1. Place the starting super-tile into the square lattice and place a regular super-tile into the square lattice so that its "U" ligand is attached to the receptor of the starting gadget.

2. Let the last placed super-tile be a (flipped) regular super-tile R; either place a (flipped) regular super-tile so that its "U" ligand is attached to the receptor of R and continue with step 4 or place a (flipped) bending super-tile such that its ligand is attached to receptor of R and continue with step 3.

3. Let the last placed super-tile be a bending super-tile B. If B is a flipped tile attach a new regular super-tile otherwise attach a flipped regular super-tile to B. The new super-tile can be attached either with "U" or "D" ligand depending on intended direction of the bend.

4. Continue with step 2 or end the structure by attaching a (flipped) terminating super-tile to the last placed (flipped) regular super-tile.

In the above procedure the super-tiles are placed into the square lattice such that they do not overlap. An example of a wave structure is depicted in Figure 4.9. As snake structures, wave structures can approximate (although more coarsely) any given shape using line segments depicted in Figure 4.10.

4.6 Proof techniques

In this section we explain the concepts and techniques we use to prove the stability of snake and wave proteins. We also introduce 2DHPSolver, a semi-automated proving tool developed to analyze the huge number of cases required for the proofs.

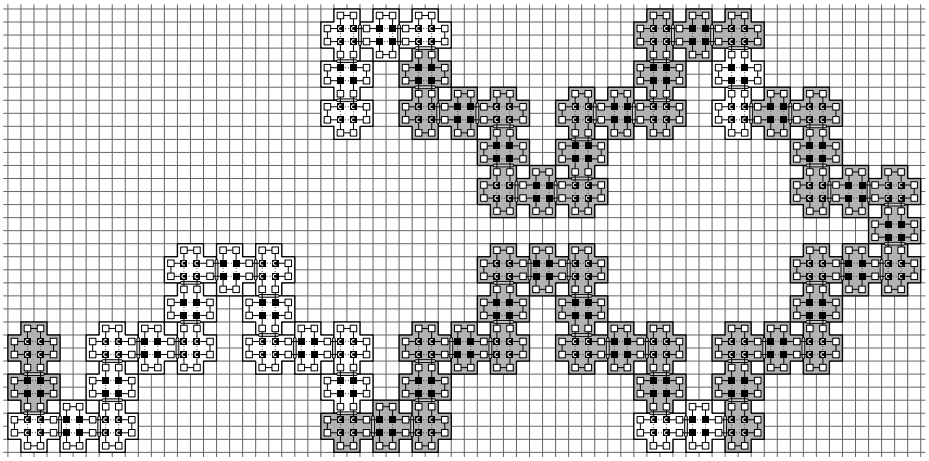


Figure 4.9: An example of a wave structure. It consists of 8 super-tiles. The borders between super-tiles are marked by the change of underlying color of the core tiles.

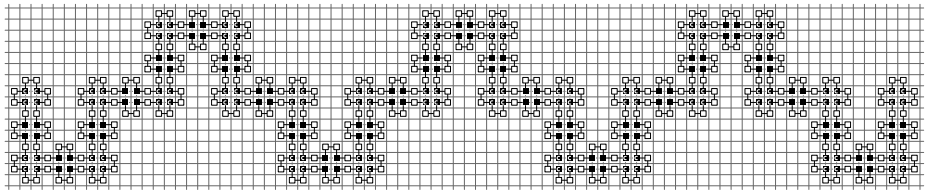


Figure 4.10: Simulation of a straight line segment with a wave structure.

4.6.1 Saturated folds in (strong) HPC model

In saturated folds under the (strong) HPC model all parts of energy function produce minimum possible values. This means: (i) every hydrophobic monomer (cysteine or non-cysteine) has two contacts with other monomers; (ii) every cysteine is involved in disulfide bridge (iii) in addition, in strong HPC model every non-cysteine hydrophobic monomer is involved in a non-cysteine bridge. Clearly, a saturated fold of a protein must be native, and furthermore, if there is a saturated fold of a protein, then all native folds of this protein must be saturated. Notice that the snake and wave structures are saturated. This property greatly simplifies the stability proof of the snake and wave structures. Before we present the proof we introduce some notations and definitions in the following paragraph that we will use in the proof.

Let F be a fold of a snake or wave protein p in 2D square lattice $\mathcal{L}$. Define a path in F as a sequence of vertices such that no vertex appears twice and any pair of consecutive vertices in the path are connected by peptide bonds. A cycle is a path whose start and end vertices are connected by a peptide bond. Note that F has only one cycle which is the entire sequence p. For $i \in \{0, 1, 2\}$, an i-vertex in the fold F is a lattice vertex (square) containing a monomer i. For instance, a square containing a cysteine monomer in F is called a 2-vertex. An H-vertex is a vertex which is either 1-vertex or 2-vertex. Define an H-path in F to be a sequence of H-vertices such that each H-vertex appears once and any pair of consecutive hydrophobic (1 or 2) monomers form an HH contact. An H-cycle in F is an H-path whose first and last vertices form an HH contact. An H-cycle of length 4 is called a core in F. Clearly, every H-path in a saturated fold is part of an H-cycle.

4.6.2 2DHPSolver: a semi-automatic prover

We have developed a tool 2DHPSolver, for proving the uniqueness of a protein design in 2D square lattice under the HP, HPC or strong HPC models. 2DHPSolver is not specifically designed to analyze the wave structures or even the constructible structures. It can be used to prove the stability of any design in 2D HP models. We use induction on the boundaries of diagonal rectangle surrounding the folds to first prove some properties of native folds and then use them to prove the uniqueness of the folds. We use 2DHPSolver in the following scenario; for any integer i, we define two *boundaries* SW_i and SE_i as the set of lattice vertices $\{[x, y]; x + y = i\}$ and $\{[x, y]; x - y = i\}$, respectively. Then we prove a property Π

(for instance, every H-monomer belongs to a core) of all native conformations of designed proteins using induction on the boundary indices. To do this, we assume that the property Π holds for the part of the conformation that lies on boundary SW_k (SE_k) for any $k < i$ (the induction hypothesis) and prove that Π holds for the part of the conformation that lies on SW_i (SE_i). In what follows we explain how 2DHPSolver can be used to assist in the proof process. More details can be found in the help document in the software package (see below for download information).

2DHPSolver has three sets of inputs: the design rules, the initial configuration, and the run-time parameters. The design rules are the set of rules that specifies the properties of the designed sequence. For instance, the subsequences that cannot be part of any designed sequence are included in the design rules (these are called *forbidden* subsequences).

2DHPSolver maintains a list of current configurations. Initially this list contains only the initial configuration which is normally just a hydrophobic monomer on a SW (SE) boundary. Each configuration that 2DHPSolver stores and processes is in fact part of a potential saturated configuration of a designed protein. The proof process is completed when the list only contains configurations that satisfy the property Π. In each iteration, one of the current configurations is replaced by all possible extensions at one square in the configuration specified by the user. Note that in displayed configurations red 1 represents a cysteine monomer, blue 1 a non-cysteine hydrophobic monomer, and uncolored 1 is hydrophobic monomer, but it is not known whether it is a cysteine or not. The following types of extensions are used in 2DHPSolver:

- extending a path;
- extending an H-path;
- coloring an uncolored H monomer.

Since a path and a 1-path can continue in 3 directions there are 6 ways to extend a path (with a 0 or 1 at each direction) and 3 ways to extend a 1-path. Furthermore, there are 2 ways to color an uncolored H monomer. For each of these possibilities, 2DHPSolver creates a new configuration which is then checked to see if it violates the rules of the design. Those which do not violate the design rules will replace the original configuration.

However, this approach will result in producing too many configurations, which makes it hard for the user to keep track of. Therefore, 2DHPSolver contains utilities for automatically finding extending sequences for each configuration which either leads to no valid

configurations, in which case the configuration is automatically removed, or to only one valid configuration, in which case the configuration is replaced by the new more-completed configuration. This process is referred to as a *self-extension*. The time required for searching for such extending sequence depends on the depth of the search, which can be specified by user through two parameters "depth" and "max-extensions". Thus, leaving the whole process of proving to 2DHPSolver by setting the parameters to high values is not practical as it could take enormous amount of time. Note that the search space is infinite, and thus cannot be searched completely automatically. Instead, the user should set run-time parameters to moderate values and use intuition in choosing the next extension point when 2DHPSolver is unable to automatically find self-extending sequences. These parameters can be changed at any time during the use of the program by the user. Figure 4.11 depicts the interface menu of the 2DHPSolver. The pseudo code algorithm of 2DHPSolver is presented in appendix A.

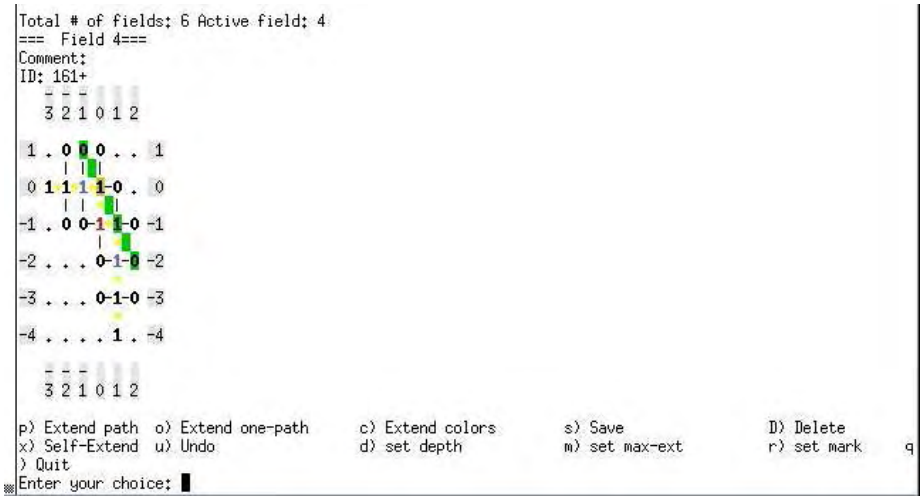
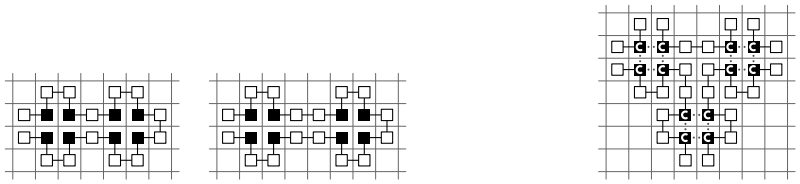


Figure 4.11: A snapshot of 2DHPSolver interface.

2DHPSolver is written in C++ and its source code is freely available under the GNU Public License (GPL). For more information and to access the source codes please visit `http://www.sfu.ca/~ahadjkho/2dhpsolver/`.

4.7 Proof of stability of the snake structures in the strong HPC model.

In this section we prove that the protein of any snake structure is stable under the strong HPC model. Let S_s be a snake structure (fold), p_s its protein and let F_s be an arbitrary native (i.e., saturated) fold of p_s.



(a) (b)

Figure 4.12: Correctly aligned cores (a) and T-aligned cores (b).

A core c is called *monochromatic* if either all its H-vertices are cysteines or all of them are non-cysteines. Let c_1 and c_2 be two cores in F_s. We say, c_1 and c_2 are adjacent if there is a path containing 0-vertices of length 2 or 3 between an H-vertex of c_1 and an H-vertex of c_2. We say c_1 and c_2 are *correctly* aligned if they are adjacent in one of the forms in Figure 4.12(a) and we say that they are T-aligned if they are two of the three cores in Figure 4.12(b). Notice that T-aligned cores can only contain cysteine monomers because otherwise we get a forbidden subsequence. The set of forbidden subsequences that we used to prove the stability of snake structures are listed in appendix B.

In what follows we prove that every H-vertex in F_s belongs to a monochromatic core and the cores are correctly aligned. We start by proving the following lemma.

Lemma 1. *Every* H*-vertex in F_s belongs to a monochromatic core and all the cores are correctly aligned, or there are three T-aligned cores and all other cores are correctly aligned.*

Proof. Let m be the largest number such that SW_i, $i < m$ does not contain any H-vertex and let n be the smallest number such that SW_i, $i > n$ does not contain any H-vertex, i.e., SW_m and SW_n are two parallel boundaries of the smallest diagonal rectangle enclosing all the H-vertices of F_s.

We prove the lemma by a two phase induction on the SW_i boundaries (cf. Figure 4.13). Let i be an integer greater than m. Assume that for any H-vertex v on SW_k where $m < k < i$,

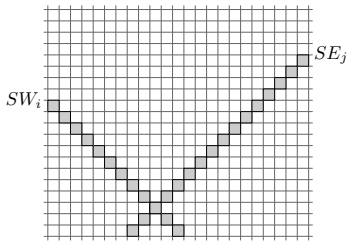


Figure 4.13: SW_i and SE_j boundaries.

v belongs to a monochromatic core c and if c is adjacent to core c' which has a H-vertex on $SW_{k'}$, $k' < i$, then c and c' are correctly aligned or T-aligned. This is in fact the induction hypothesis. Now we prove the following claim which is the induction conclusion.

Claim 1. *If there is an* H*-vertex w on SW_i, then*

(1) w is on a monochromatic core c; and

(2) if c is adjacent to core c' which has an H*-vertex, on SW_j, $j < i$, then c and c' are either correctly aligned or T-aligned.*

Proof. We show that if (1) or (2) does not hold for w then the configuration violates at least one the properties of the snake structures (for instance, it contains a forbidden subsequence or it is not a saturated fold). This is done by enumerative case analysis of all possible extensions of this configuration and showing that each branch will end in a configuration that violates one of the properties of the snake structures.

This process requires the analysis of many configurations which is very hard and time consuming to do manually. Therefore, we used 2DHPSolver to assist in analyzing the resulting configurations. The program generated proof of this step of the induction can be found on our website at `http://www.sfu.ca/~ahadjkho/2dhpsolver/snake.htm`. Please be advised that this is a PDF document containing 2707 pages and 16543 images.

□

If Claim 1 holds for any $i > m$ in a such way that c and c' are correctly aligned then the proof of Lemma 1 is complete. Therefore, assume that there exists an integer $i > m$ such that the cores c and c' in Claim 1 are T-aligned. We stop the induction here and start the second induction from the opposite direction. Due to the symmetry there exists an integer $j < n$ such that the two possibly new cores d and d' are T-aligned while all H-vertices on any

SW_k where $k > j$ belong to correctly aligned cores. Notice that a T-aligned configuration introduce two occurrences of the subsequence $es = (020)^4$ (cf. Figure 4.12(b)), and since p_s contains exactly two occurrences of e therefore there can be at most one T-aligned configuration in F_s. It follows that $i = j$ which completes the proof of Lemma 1.

□

Lemma 2. *Every* H*-vertex in F_s belongs to a monochromatic core and all the cores are correctly aligned.*

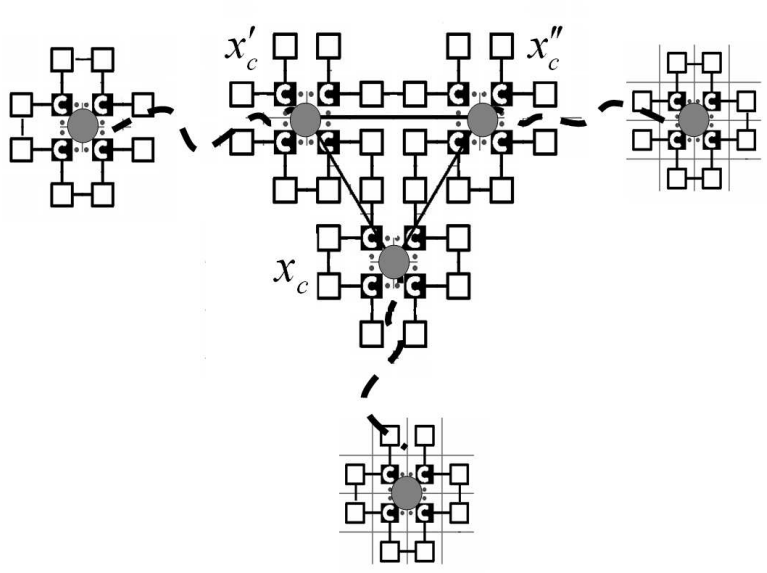


Figure 4.14: The corresponding graph of a saturated fold of a snake structure with three t-aligned cores c, c', and c''.

Proof. By Lemma 1, every H-vertex is in a core. Consider a graph G defined as follows. For every core c of F_s, let x_c be a vertex in G. Furthermore, two vertices x_c, and $x_{c'}$ are connected in G if and only if cores c and c' are adjacent in F_s. We show that G is acyclic. To the contrary, assume that C is a cycle in G. It is easy to see that if all the cores corresponding to vertices of C in F_s are correctly aligned, we get a closed subsequence which is not the entire p_s, a contradiction. Thus C contains vertex x_c corresponding to a core c which is T-aligned with two other cores c' and c'' (cf. Figure 4.12(b)). Note that, by

Lemma 1, all other cores in F_s are correctly aligned. Since c is connected to at least one core other than c' and c'', vertex x_c has degree at least three in G. If C is of length more than three then it can only contain one of the cores c' and c'' and the rest of its cores are correctly aligned. However, in this case we also get a close subsequence which is not the entire p_s. Thus C only contains the three vertices x_c, x'_c and x''_c. Furthermore G cannot have any cycle other than C because there is at most one T-aligned configuration in F_s. Since x_c, x'_c and x''_c are all of degree at least 3 and there is only one cycle in G, there are at least 3 vertices of degree 1 in G (cf. Figure 4.14). These vertices correspond to cores in F_s that contain an occurrence of $e = (\mathsf{020})^4$, which implies that F_s contains at least 5 occurrences of e, a contradiction. It follows that, G is acyclic.

□

Theorem 1. *Let F_s be an arbitrary saturated fold of a snake protein p_s that folds into a snake structure S. $F_s = S_s$ (i.e, p_s is stable).*

Proof. By Lemma 2 every H-vertex in F_s belongs to a monochromatic core and all the cores are correctly aligned. Construct graph G from the cores in F_s as described in the proof of Lemma 2. By a similar argument used in Lemma 2 G is acyclic. Furthermore G does not have any vertex of degree more than 2 as otherwise there would be at least three vertices of degree 1 in G and hence at least three occurrences of e in F_s, a contradiction. Thus G is a path which implies that F_s is a linear constructible structure. Now the first core c_1 in F_s (c_1 is adjacent to exactly one core) corresponds to t_1 of S_s. By comparing the sequence of p_s in core c_i of F_s and core t_i of S, for $i > 1$, it follows by induction that F_s has the same structure as S_s. Thus, p_s is stable. □

4.8 Stability of the wave structures in the HPC model

In this section we prove that the wave structures are stable in the HPC model. Let S_w be a wave structure, p_w its protein and let F_w be an arbitrary native (i.e., saturated) fold of p_w.

Similar to the proof of the stability of snake structures we first prove that every H-vertex in F_w belongs to a monochromatic core and the cores are correctly aligned. The set of forbidden subsequences that we used to prove the stability of wave structures are listed in appendix C.

Lemma 3. *Every* H*-vertex in F_w belongs to a monochromatic core and either all the cores are correctly aligned or there are three cores in F_w which are T-aligned while all other cores are correctly aligned.*

Proof. We start by proving the following claim.

Claim 2. *Every* H*-vertex in F_w belongs to a monochromatic core.*

Proof. We prove the claim by induction on SW_i. More specifically we prove that for every i and every H-vertex v on SW_i, v is in a monochromatic core. For the base case, consider largest m such that for any $j < m$, there is no H-vertex on SW_j. Then the claim is trivially true. For the induction step, assume that the claim is true for every H-vertex on SW_k for any $k < i$, it is enough to show that the claim is true for any H-vertex on SW_i. Similar to Claim 1, we use the 2DHPSolver to prove the induction step.

`http://www.sfu.ca/~ahadjkho/2dhpsolver/core-monochromatic-proof`.

□

Next, we prove the following claim using the 2DHPSolver tool.

Claim 3. *Let c_1 and c_2 be two adjacent monochromatic cores in F_w. Then c_1 and c_2 are either aligned correctly or T-aligned.*

The program generated proof of this claim can be found on our website at
`http://www.sfu.ca/~ahadjkho/2dhpsolver/core-alignment-proof`.

□

The main result follows from Lemma 3, Lemma 2 and a similar argument to the proof of Theorem 1.

Theorem 2. *Every* H*-vertex in F_w belongs to a monochromatic core and all the cores are correctly aligned. Hence, $F_w = S_w$, i.e., all wave structures are stable.*

Chapter 5

Structure approximating IPF in 3D hexagonal lattice

5.1 Introduction

In this chapter we extend the IPF design and introduce the first robust class of protein design in a 3D setting. We use the 3D hexagonal prism lattice as the underlying design lattice. The hexagonal prism lattice is composed by stacking horizontal hexagonal grids ("honeycomb nets") on top of each other, cf. Figure 5.1. Two facts about this lattice are useful to us in our construction. First these lattices have a relatively low degree (the number of neighbors of a vertex) of 5. The cubic lattice, for example, had a degree of 6. This lower degree simplifies our designs. At the same time, we show that relative to its degree this lattice is remarkably good at representing a large class of natural protein structures.

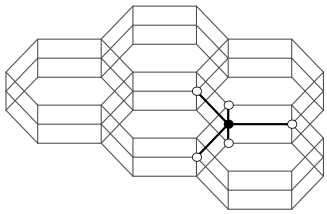


Figure 5.1: An example of hexagonal prism lattice.

We design a class of structures (called *tubular structures*) and corresponding proteins in

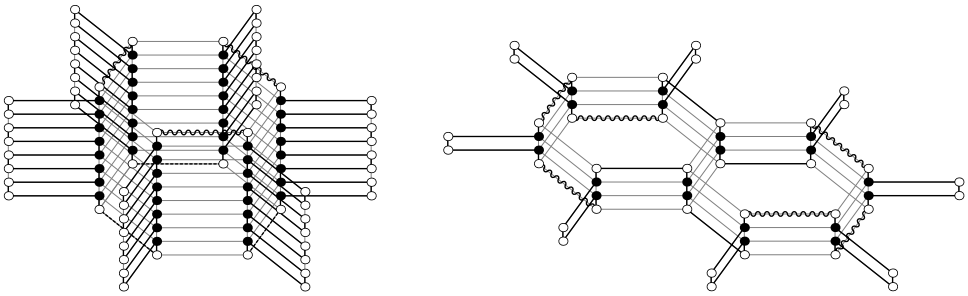


Figure 5.2: (a) Illustration of a tube with a hydrophobic core of height 8 — the wavy lines at the top and dashed lines at the bottom represents loops. (b) Illustration of a connector.

the 3D hexagonal prism lattice and show that each protein folds into the corresponding tubular structure. The building blocks of tubular structures are *tubes* and *connectors* shown in Figure 5.2(a) and (b), respectively, where hydrophobic monomers (cysteine or non-cysteine) are depicted with black beads and polar ones with white beads. The hydrophobic core of the connector consists of 2 layers of two adjacent hexagons. The connector can be attached to 4 tubes (one per top/bottom of each hexagon). An example, with 3 tubes attached to the connector is shown in Figure 5.4. Such design is sufficiently robust to roughly approximate any given shape.

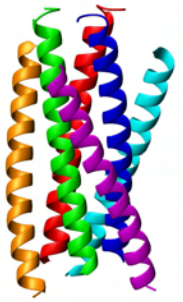


Figure 5.3: A coiled coil structure formed by 6 alpha-helices in protein gp41. Figure is taken from wikipedia (`http://en.wikipedia.org/wiki/Image:Gp41_coiled_coil_hexamer_1aik_sideview.png`) used by permission of WillowW.

Interestingly, a tube consists of six parallel "alpha helix"-like structures. Similar designs appear in nature as a *coiled coil* structural motif in which 2–6 alpha-helices are coiled together, cf. Figure 5.3. Many coiled coil type proteins are involved in important biological functions such as the regulation of gene expression e.g. transcription factors [165, 109].

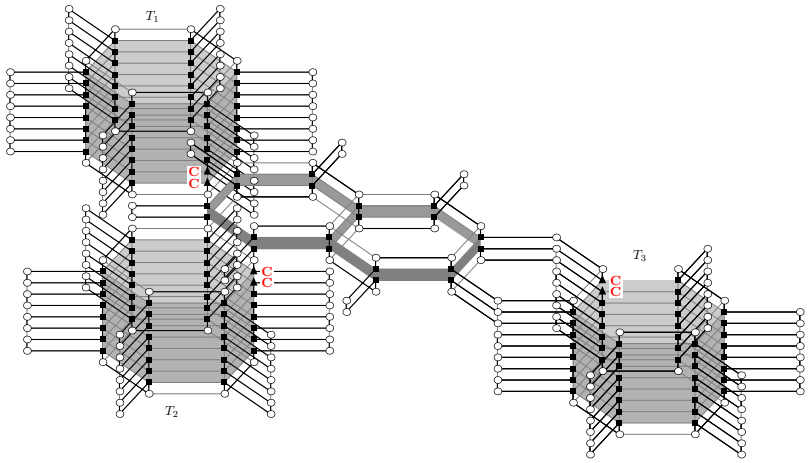


Figure 5.4: An example of a tubular structure showing the ability to branch (on the left). Polar, hydrophobic and cysteine monomers are depicted as empty circles, squares and triangles, respectively. Hydrophobic cores of 3 tubes and a connector are highlighted.

We show that a tubular structure is one of the native folds of its protein under HPC model. We conjecture that the proteins of the tubular structures are structurally stable, i.e., all the native folds of a protein from tubular structures are completely similar to each other.

We are able to prove this formally for infinite subclass of the simple structures (consisting of one connector and three tubes, cf. Figure 5.4) under the assumption that each of three tubes is sufficiently long. In addition, we assume that our proteins are closed chains of monomers, a similar assumption as used in [1], i.e., that the beginning and the end of the sequence are adjacent in the lattice. Note that tubular structures from this subclass are not stable under the HP model, thus our results show that using disulfide bridges in our designs helps to stabilize them.

Despite the tremendous amount of work on protein design for 2D lattices, as far as we know, this is the first general design of arbitrary long stable proteins for the 3D lattice. Given that 3D is the realistic setting, we believe that this work could eventually help in designing proteins with applications to drug design and nanotechnology.

5.2 Preliminaries

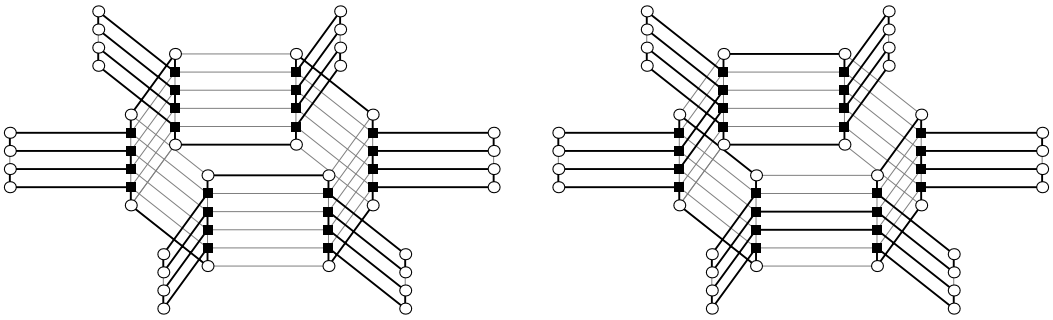
In this section we will review the 3D hexagonal HPC model and introduce some terminology used in the chapter.

5.2.1 3D Hexagonal HPC model

The HPC model in 3D hexagonal lattice is a straight forward extension of the HPC model in 2D square lattice we defined in section 4.3. A protein in HPC model is represented as a string $p = p_1 p_2 \dots p_{|p|}$ in $\{0, 1, 2\}^*$, where "0" represents a polar monomer (depicted in figures as empty circles), "1" a hydrophobic-non-cysteine (depicted as black squares) and "2" a cysteine monomer (depicted as black triangles). We use H to represent a monomer which could be either 1 or 2 (depicted in figures as a black circle). The proteins are folded onto the regular lattice. A *fold* of a protein p is embedding of a path of length n into the lattice.

The vertices adjacent to a vertex are called the neighbors of that vertex. As depicted in Figure 5.1(a), each vertex has 5 neighbors: 3 horizontal neighbors lying in the same hexagonal grid and 2 vertical neighbors lying above and below the vertex in the parallel hexagonal grids.

A protein will fold into a conformation with the minimum free energy, also called a *native fold*. The energy function in the HPC model consists of two parts: *hydrophobic interactions* and *disulfide bridges*. The hydrophobic monomers which are not consecutive in the protein but are adjacent in the lattice form *contacts*. Each contact contributes -1 to the total energy. The cysteines act as hydrophobic monomers for this part of energy function. In addition to hydrophobic interactions a pair of cysteines which are not consecutive in the protein but are adjacent in the lattice form disulfide bridges and further reduce the energy of the fold. Unlike the hydrophobic interactions in which a hydrophobic monomer can take part in several contacts, a cysteine can only participate in one disulfide bridge. Therefore, the number of disulfide bridges contributing to the energy of a fold is equal to the number of pairs in the maximum matching in the graph of potential disulfide bridges. Each disulfide bridge contributes -1 to the total energy. Hence, a fold with the lowest free energy corresponds to a fold with the largest number of HH contacts and disulfide bridges.



(a) (b)

Figure 5.5: Two native folds of the substring $t = (0100110010)^6$. These two folds are similar.

5.2.2 Structural stability

Every protein and its fold define a mapping from the lattice vertices to the set $\{0, 1, 2, W\}$, where W represents "water" or an empty unoccupied position. We say that two folds of the same protein are *similar* if they define the same mapping. If all native folds of a given protein are similar to each other, then the protein is called *structurally stable*. Note that all native folds of a structurally stable protein have exactly the same shape (from outside they appear as the same fold). For instance, the string $t = (0100110010)^6$ is structurally stable, but not stable. Figure 5.5 depicts both native folds of this string. It is easy to see that the mappings defined by t and its two folds are identical, i.e., the folds are similar.

5.2.3 Terminology

A lattice vertex containing an $X \in \{0, 1, 2\}$ monomer is called an X-vertex. An H-vertex is either a 1-vertex or a 2-vertex. A neighbor of a vertex v which is an X-vertex is called X-neighbor.

Consider a fold F. A path in F is a sequence of vertices $(x_1, x_2, \ldots, x_k)$ such that consecutive vertices are connected by peptide bonds. We say that F contains an occurrence of substring $w_1, w_2, \ldots, w_k$ if there is a path $(x_1, x_2, \ldots, x_k)$ in F such that x_i is a w_i-vertex. We number hexagonal grids of the lattice (also referred to as *planes*) with integer numbers, and denote the i-th grid by H_i. Consider vertex $x \in H_i$. We denote the vertical neighbor of x in H_{i+1} (above x) by x^1, and recursively, the vertical neighbor of x^j in H_{i+j+1} by x^{j+1}. Similarly, we denote the neighbor of x in H_{i-1} by x^{-1}, and the neighbor of x^{-j} in H_{i-j-1} by x^{-j-1}.

Let G_x be the graph of all H-vertices in H_i which are reachable from x by a path of H-vertices in H_i. Let G be a set of vertices in H_i. Then for $j \geq 1$, let G^j be the graph of all vertices in H_{i+j} which have a neighbor in G^{j-1}, and G^{-j} be the graph of all vertices in H_{i-j} which have a neighbor in G^{-j+1}, i.e., G^j and G^{-1}, $j \neq 0$, are vertical copies of the set G. Note that G_x is a planar graph (as H_i is as well). The degree of a vertex in G_x is called a *plane degree*. Let B_x be the boundary cycle of G_x, i.e., the set of vertices of G_x which lie on the outer face of G_x. A *component* in a fold F is a maximal set of H-vertices for which there is a path of H-vertices between any pair of them.

Let C be a component that lies in the planes H_{j+1} to H_{j+r}. Let layer C_i be a graph of all vertices of C in plane H_{j+i+1}. We say that projections of layers C_i and C_k are the same (C_i is subset of C_k) if $C_i^{k-i} = C_k$ ($C_i^{k-i} \subseteq C_k$). In this case we write $C_i \simeq C_k$ ($C_i \stackrel{\sim}{\subset} C_k$).

The plane containing C_i will be denoted by $H(C_i)$.

5.2.4 Saturated folds in 3D hexagonal HPC model

Similar to the proteins we used in our 2D design, the proteins we use in this chapter are saturated, i.e, the number of possible contacts and disulfide bridges of their native folds is maximal with respect to the number of hydrophobic "1" and cysteine "2" monomers contained in the protein. The following useful observation characterizes native folds of such proteins.

Observation 1 (Saturated folds)**.** *Let $p \in 0\{0,1,2\}^*0$ be a protein, and F be the fold of p. If for every H-vertex v, three out of five edges incident with v are contacts and in addition if v is a cysteine it belongs to a maximum matching in the graph of potential disulfide bridges, then (a) F is a native fold of p; and (b) any other native fold of p satisfies these properties. We will call a fold satisfying these properties a* saturated *fold.*

The proof of the observation follows by a simple argument that any hydrophobic vertex v can have at most three contacts since it is connected to exactly two neighbors with a peptide bond and furthermore any cysteine monomer can be involved in at most one disulfide bridge. Note that not every protein has a saturated fold.

5.3 Tubular structures and their proteins

The first basic building block of our *tubular structures* is a *tube*, depicted in Figure 5.2(a). A tube consists of 6 identical "alpha helix"-like subfolds of the substring $p_n = (\mathsf{H00H})^n$ forming a $2 \times 2n$ vertical zig-zag pattern ("plate").

The plates are connected to each other with 6 short loops (3 at the top and 3 at the bottom), each consisting of only two polar monomers. Thus, the hydrophobic core is completely surrounded by polar monomers, i.e., the fold is saturated. The complete protein string for the tube is $t_n = (\mathsf{0}p_n\mathsf{0})^6$. We assign the first and the second H monomer of one of the plates of each tube to cysteine monomers $\mathsf{2}$. We represent the fold of t_n by $\mathcal{T}_n$. The height of the hydrophobic core of the tube $\mathcal{T}_n$ is $2n$.

The second building block of our tubular structures is a *connector*, depicted in Figure 5.2(b). A connector can be formed by overlapping two very short tubes (with height of hydrophobic core 2). Two tubes or a tube and a connector can be connected to one protein structure in two ways as follows. First, one top loop of the first tube is overlapped with a bottom loop of the second tube/connector, vice versa, and the peptide bonds between two polar monomers of each loop are disconnected. This way of connecting two components is called *vertical* connection. Tubes T_1 and T_2 in Figure 5.4 are vertically connected to the connector. In the second way, called *horizontal* connection, the tubes or the tube and the connector are placed beside each other such that they have H-vertices in exactly one common plane H_i and exactly two H-vertices of the first component are connected to two H-vertices of the other component each through one $\mathsf{0}$-vertex. Tube T_3 in Figure 5.4 is horizontally attached to the connector. The class of *tubular structures* contains all structures formed by connecting tubes and connectors (such that no space violation occurs). We choose to vertically or horizontally connect a tube to a component in a tubular structure such that no pair of H-vertices in the same plane and in middle layers of different tubes are at distant three of each other. Since, there is no substring $\mathsf{000}$ in the protein of any tubular structure, this condition ensures that the tubes in a tubular structures do not directly connect to each other through the H-vertices in their middle layers. This will greatly simplifies the stability proof of the structures.

Since, the folds of tubular structures are saturated, by Observation 1, they are native folds to corresponding proteins (which can be easily reconstructed from the folds).

5.4 Stability of tubular structures

In what follows we will show that the proteins of a basic class of tubular structures $\mathcal{C}$ are structurally stable. The structures in class $\mathcal{C}$ are built from one connector and three tubes, cf. Figure 5.4. We will assume that three tubes $\mathcal{T}_{k_1}, \mathcal{T}_{k_2}, \mathcal{T}_{k_3}$ used to construct these structures are sufficiently long. In particular, we will assume that $k_1, k_2, k_3 \geq 712$. We conjecture that this structure is structurally stable also for other values of k_1, k_2, k_3 and that all tubular structures are structurally stable. Let q be a protein string of a structure in $\mathcal{C}$ and Q be its original fold.

Definition 1 (sparse protein)**.** We say that a protein is *sparse* if does not contain HHH as a substring and does not start or end with H.

5.4.1 Types of H-vertices

Let F be a saturated fold of a sparse protein. Then each H-vertex has exactly three contacts, i.e., it has at least three H-neighbors and the remaining two neighbors are connected (via a peptide bond) and at most one of the two is an H-vertex. We can classify every H-vertex x of F to one of the five types based on the position of its 0-neighbor(s), cf. Figure 5.6:

(a) vh-type: x has one vertical 0-neighbor (on top or below) and one horizontal 0-neighbor (in the same hexagonal grid);

(b) vv-type: x has two vertical 0-neighbors;

(c) hh-type: x has two horizontal 0-neighbors;

(d) h-type: x has one horizontal 0-neighbor;

(e) v-type: x has one vertical 0-neighbor.

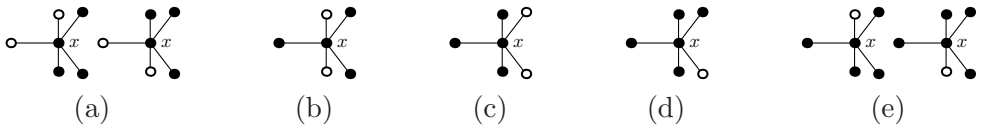


Figure 5.6: Five types of possible neighborhood of an H-vertex x: S-vertices: (a) vh, (b) vv, (c) hh; and D-vertices: (d) h and (e) v. For S-vertices x is connected to two 0 neighbors by peptide bonds, while for D-vertices x is connected to the 0-neighbor and one of the H-neighbors.

For every $\mathsf{X} \in \{\mathsf{vv}, \mathsf{hh}, \mathsf{h}, \mathsf{v}\}$ an H-vertex of type X, will be called X-vertex. Furthermore, any H-vertex with two 0-neighbors is called a S-*vertex* and an H-vertex with one 0-neighbor is called a D-*vertex*.

Definition 2 (connections)**.** Let $u, v \in \{0, 1, 2, \mathsf{H}, \mathsf{S}, \mathsf{D}, \mathsf{vv}, \mathsf{hh}, \mathsf{h}, \mathsf{v}\}$ and $s \in \{0, 1, 2, \mathsf{H}\}^+$. We say that two vertices x and y are *s-connected* if there is a path $x, v_1, v_2, \ldots, v_k, y$ in the lattice such that v_i is an s_i-vertex. If x is a u-vertex and y is a v-vertex, this path is called an *usv-connection*. If the end points x and y are H-vertices and belong to two different components, we say that these components are *usv-connected*. If $s = \mathsf{00}$ and $u, v \neq 0$, we will shorten this notation as $(u \asymp v)$-connection. In particular, we will be interested in H0H-connections and $(\mathsf{S} \asymp \mathsf{h})$-connections.

A *usv*-connection with end points x and y is called *internal*, if x and y are in the same component, and otherwise it is called *external*. We say that two *usv*-connections with end points at x, y and x', y', respectively, are *parallel* if x (y) is directly above/below x' (y'), i.e., $x' = x^i$ and $y' = y^j$, for some integers i, j, and all vertices between x and x' (y and y') are H-vertices. Note that it is also possible that x and y' are u-vertices and x' and y are v-vertices.

We have the following observations:

Observation 2. *Let F be an arbitrary saturated fold of q. Then F contains 6* H0H*-connections, 52* S*-vertices, the number of* D*-vertices is 4 modulo 6 and it contains* 36 $(\mathsf{S} \asymp \mathsf{D})$*-connections. F does not contain* HHH*,* 000*,* H0H0H *and* H0HH*, but it does contain one occurrence of* 20100101*.*

Observation 3. *Let F be a saturated fold of a sparse protein. Then every* H*-vertex of F is either a* vh*-vertex,* vv*-vertex,* hh*-vertex,* h*-vertex or* v*-vertex. Furthermore, any neighboring* 0*-vertex and* H*-vertex are connected by a peptide bond.*

Claim 4. *Let F be a saturated fold of a sparse protein with no* H0HH *as a substring. Then no* v*-vertex in F can connect directly to an* h*-vertex.*

Proof. Consider a v-vertex x. Without loss of generality assume that its 0-neighbor is x^1. Assume to the contrary that x connects to an h-vertex. Two cases are possible: first, x connects to its horizontal h-neighbor z cf., Figure 5.7(a). Then z^1, x^1, x, z form the substring H0HH, a contradiction. Second, x connects x^{-1} which is an h-vertex. Let z be the

horizontal 0-neighbor of x^{-1}. Then z^1, z, x^{-1}, x form the substring H0HH, a contradiction cf., Figure 5.7(b). □

Claim 5. *Let F be a saturated fold of a sparse protein. No* v*-vertex can connect to an* h*-vertex via two* 0*-vertices.*

Proof. Consider a v-vertex x. Without loss of generality assume that its 0-neighbor is x^1. Assume to the contrary that x^1 connects to an h-vertex via one 0-vertex y. If y is a horizontal neighbor of x^1 then it would connect down to a horizontal neighbor of x which is not an h-vertex. Hence, $y = x^2$. Furthermore, x^2 should connect to an h-vertex, hence it cannot connect to x^3. Therefore it must connect to one of its horizontal neighbor z. Since, z is an h-vertex, z^{-1} is an H-vertex. However, this a contradiction, as x^1 would have to connect to three vertices: x, x^2 and z^{-1} Figure 5.7(c). □

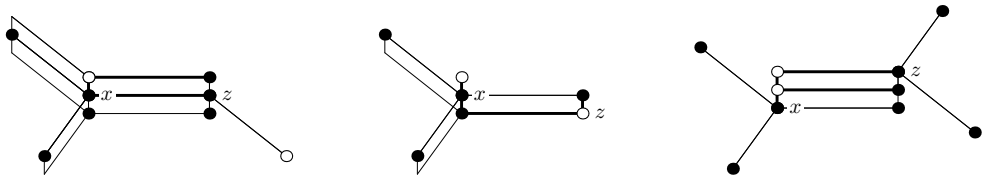


Figure 5.7: Case analysis showing that a vh-vertex cannot directly (a) and (b); or via two 0-vertices (c) connect to an h-vertex

The above two claims imply the following lemma.

Lemma 4. *Let F be a saturated fold of a sparse protein with no* H0HH *as a substring. Any occurrence of substring* $(00HH)^k$ *in F contains either only* v*-vertices or only* h*-vertices.*

5.4.2 Types of components

In this section we study all possible components that can arise in saturated folds of q. We first classify all components to three categories and then study which of these can appear in saturated folds of q.

Let F be a saturated fold of a sparse protein and C a component in F. Assume that C lies in the planes $H_s, \ldots, H_e$. Note that any H-vertex of plane degree one in the first or last layer of C is adjacent to at least three 0-vertices, a contradiction. Hence, we have the following observation.

Observation 4. *Let F be a saturated fold of a sparse protein and let C be a component in F. Then all vertices of the first or last layer of C have plane degree* 2 *or* 3.

The following definition defines several types of components.

Definition 3 (tube, simple tube, 2-layer component, wall, and complex component)**.** A *tube* is a component such that all its layers are identical and each layer contains only vertices of plane degree two (a cycle). A *simple tube* is a tube with only one hexagon in each layer. A *2-layer component* is a component with two identical layers which have no vertex with plane degree 1 and at least one vertex with plane degree 3. A *wall* is a component such that all its layers are identical and each layer is a single path. Finally, a *complex component* is a component C such that there is some i for which C_i and C_{i+1} are different.

We have the following observations.

Observation 5. *Any component C in a saturated fold of a sparse protein is one of the following three types: a tube, a 2-layer component or a complex component.*

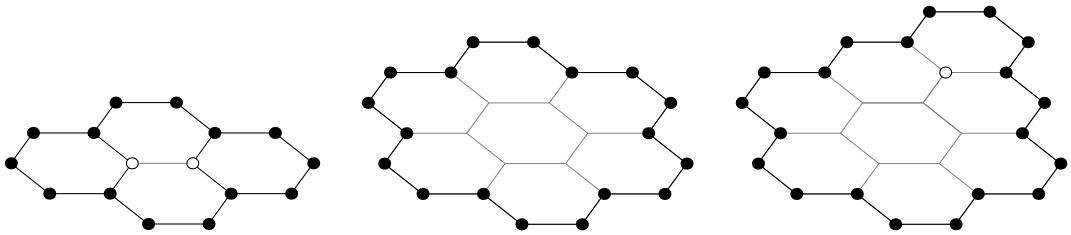


Figure 5.8: One layer of (a) the smallest non-simple tube; (b) the smallest non-simple tube without occurrences of H0H; and (c) the smallest non-simple tube with one occurrence of H0H per layer.

Observation 6. *Let F be a saturated fold of a sparse protein. If F contains a tube then the height (number of layers) of this tube is at least* 2*. One layer of the smallest non-simple tube is depicted in Figure 5.8(a). It contains two occurrences of* H0H *per layer, i.e., at least* 4 *such occurrences. One layer of the smallest non-simple tube with no occurrences of* H0H *is depicted in Figure 5.8(b). One layer of the smallest tube with one occurrence of* H0H *per layer is depicted in Figure 5.8(c).*

5.4.3 Different types of complex components

In what follows we further classify different types of complex components which can occur in saturated folds of sparse proteins with at most six occurrences of substring H0H.

Complex components with a vv-vertex

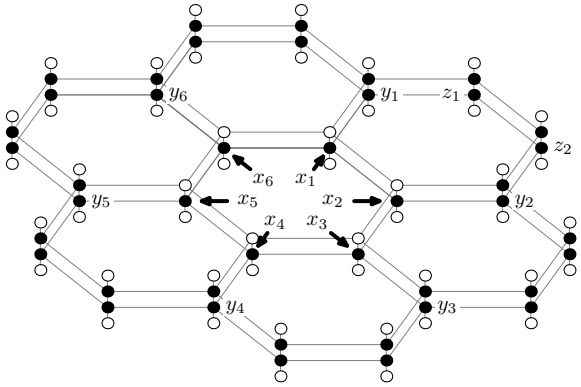


Figure 5.9: Part of a complex component with a vv-vertex. The arrows are pointing at six vv-vertices.

Lemma 5. *Let F be a saturated fold of a sparse protein with no occurrences of substrings* H0HH *and* H0H0H *and at most six occurrences of substring* H0H*. Consider a complex component C of F containing a* vv*-vertex. Then C has 6* vv*-vertices forming a hexagon, lies in two layers which are almost identical, except for the six* vv*-vertices which are replaced with* 0*-vertices in the other layer, and neither layer contains a vertex of plane degree* 1*. We will call such a complex component, a* vv-component*. A* vv*-component contains 6 occurrences of* H0H*.*

Proof. Any vv-vertex must be adjacent to at least two other vv-vertices in its plane, otherwise, either there is a 0-vertex connected to three H-vertices (with a peptide bond), or we get a substring H0H0H which cannot occur in F. Therefore, any set of vv-vertices in a plane forms a graph with no vertices of plane degree 1. Each vv-vertex on the boundary of this graph is adjacent to one non-vv-vertex which creates a distinct H0H substring. Since there are only 6 occurrences of H0H in F, the boundary of this graph must be a hexagon, i.e., C contains exactly 6 vv-vertices $x_1, \ldots, x_6$ located on a single hexagon, cf. Figure 5.9. Furthermore, C does not contain a vertex with plane degree 1. Assume to the contrary that v is a vertex with plane degree 1 and let k be the smallest number such that v^k is a

vertex with plane degree more than 1 (note that such a k exists). Let w be a horizontal H-neighbor of v^k. Now, the path (w, w^{-1}, v^{k-1}) is an H0H-connection which is different from the H0H-connections containing the vv-vertex of F, a contradiction.

For $i = 1, \ldots, 6$, let y_i be the non-vv horizontal neighbor of x_i. Consider y_1. One of its vertical neighbor is an H-vertex while the other is a 0-vertex, cf. Figure 5.9. Without loss of generality assume y_1^1 is an H-vertex. Let z_1 be the horizontal neighbor of y_1 which is closer to y_2. Since C does not contain any vertex of plane degree 1, all the horizontal neighbors of y_1^1 except x_1^1, are H-vertices. In addition, y_1^2 must be a 0-vertex otherwise, F would contain the substring H0HH, a contradiction. It follows that z_1 is an H-vertex and z_1^{-1} and z_1^2 are 0-vertices otherwise, we get additional H0H-connections, a contradiction.

Next, we show that y_2^1 is an H-vertex. Let z_2 be the common neighbor of z_1 and y_2. Clearly, z_2 is an H-vertex otherwise we get another H0H-connection, a contradiction. Similarly, z_2^{-1} and z_2^2 are 0-vertices and z_2^1 is an H-vertex. It follows that y_2^1 is an H-vertex. By similar arguments, we can show that for every $i = 1, \ldots, 6$, y_i^1 is an H-vertex and y_i^{-1} and y_i^2 are 0-vertices. Since there is no other occurrence of H0H in F, it is easy to see that the whole component lies in two layers (the layers containing y_i's and y_i^1's) which are almost identical with exception that 6 vv-vertices in lower layer replaced with 0-vertices in the upper layer. □

Note that a vv-component is essentially a 2-layer component which is missing vertices of one hexagon in one of the two layers.

Complex components without a vv-vertex

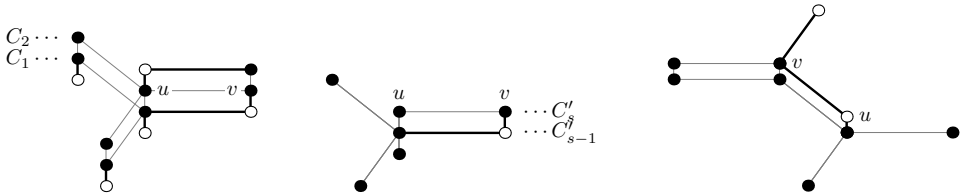


Figure 5.10: Analysis of a complex component without a vv-vertex: (a) the case in which $C'_2 \neq C_1$; (b) the case in which C'_i is not a subset of C_1; (c) the case when C'_s is not a subset of $V^{2,2,2}$.

Lemma 6. *Let F be a saturated fold of a sparse protein with no* H0HH *as a substring. Let C be a complex component of F without a* vv*-vertex and $C_1, \ldots, C_r$ its layers. Let $V^{2,2,2}$ be*

the set of all H*-vertices in* F *with plane degree* 2 *such that both its horizontal* H*-neighbors have plane degree* 2 *as well.*

(a) For $k \geq 1$*, let* C'_k *be a subset of* C_k *consisting of components of* C_k *which are intersecting projection of* C_1*. Let* s *be the smallest integer such that layer* C'_s *is different from* C_1*. Then* $s > 2$ *and* C'_s *is a collection of paths where each path is a subset of* $C_1 \cap V^{2,2,2}$*.*

(b) For $k \leq r$*, let* C''_k *be a subset of* C_k *consisting of components of* C_k *which are intersecting projection of* C_r*. Let* e *be the largest integer such that layer* C''_e *is different from* C_r*. Then* $e < r - 1$ *and* C''_e *is a collection of paths where each path is a subset of* $C_r \cap V^{2,2,2}$*.*

Proof. We prove only part (a) of the lemma, part (b) follows by symmetry. Since there is no vv-vertex in C, C_2 and hence, also C'_2 is a superset of the C_1. We show that these two layers are identical. To the contrary assume that C'_2 contains a vertex w such that its vertical neighbor in the plane $H(C_1)$ is a 0-vertex. Since, C'_2 is intersecting projection C_1, there must be a shortest path connecting w to some vertex u of C^1_1. Note that $u \in C'_2$ and $u^{-1} \in C_1$. Let v be the neighbor of u on this paths, i.e., v is an H-vertex in C_2 and v^{-1} is a 0-vertex. Since, the plane degree of u^{-1} is at least 2, its horizontal neighbors other than v^{-1} are H-vertices. Since, $C_1 \overset{\sim}{\subset} C'_2$, all horizontal neighbors of u are H-vertices, i.e., u is a v-vertex. Therefore, u^1 is a 0-vertex. Furthermore, since there is no vv-vertex in F, v^1 is an H-vertex, cf. Figure 5.10(a). Since, u is a D-vertex, it is connected to one of its H-neighbors, say z. Then, v^1, u^1, u, z form the substring H0HH, a contradiction. Hence, $C_1 = C'_2$.

Let s be the smallest integer such that C'_s is different from C_1. Since $C_1 \simeq C'_2$, it follows that $s > 2$. Next, we show that C'_s is a subset of $C_1 \simeq C'_{s-1}$. Assume the contrary. Since C'_s is intersecting projection of $C_1 \simeq C'_{s-1}$, there exists an H-vertex v and its horizontal H-neighbor u in C'_s such that v^{-1} is a 0-vertex and u^{-1} is a H-vertex in C'_{s-1}. Since $C'_{s-1} \simeq C'_{s-2} \simeq C_1$, the plane degree of u^{-1} is 2 and u^{-2} is an H-vertex, cf. Figure 5.10(b). Hence, u^{-1} is a D-vertex, i.e., it is connected to some H-vertex z. Then v, v^{-1}, u^{-1}, z form the substring H0HH, a contradiction.

Finally, note that any vertex with plane degree 3 in C'_{s-1} must have a 0-neighbor in the plane $H(C_s)$, as otherwise it would have five H-neighbors. Since, C'_s is a subset of $C_1 \cap V^2$, where V^2 is the set of all H-vertices in F with plane degree two, C'_s is a collection of paths.

Finally, let us prove that each path in C'_s lies in $V^{2,2,2}$. Assume the contrary. Then the end point v of such a path in C'_s has a 0-neighbor u such that u^{-1} is an H-vertex of plane degree 3 in C'_{s-1}. Hence, u^{-1} is a v-vertex and we have an occurrence of H0HH (cf. Figure 5.10(c)), a contradiction. □

Lemma 7. *Let F be a saturated fold of a sparse protein with no occurrences of the substring* H0HH*, and at most six occurrences of the substring* H0H*. Let C be a complex component of F without a* vv*-vertex and $C_1, \ldots, C_r$ be its layers. Let $\bar{s} > 2$ ($\bar{e} < r - 1$) be the smallest (largest) integer such that $C_{\bar{s}}$ ($C_{\bar{e}}$) is different from C_1 (C_r). Then both $C_{\bar{s}}$ and $C_{\bar{e}}$ contain a single path, and each of the layers $C_1, \ldots, C_{\bar{s}}, C_{\bar{e}}, \ldots, C_r$ is connected. Furthermore, each complex component creates at least four occurrences of the substring* H0H *in F, two between layers $C_{\bar{s}-1}$ and $C_{\bar{s}}$ and other two between layers $C_{\bar{e}}$ and $C_{\bar{e}+1}$.*

Proof. Let C'_k, C''_k be the sets and s, e the integers defined in Lemma 6. By this lemma, both C'_s and C''_e are collections of paths. Each path in C'_s and C''_e creates two new occurrences of substring H0H. Therefore, the total number of paths in C'_s and C''_e is either 2 or 3.

First, assume that C'_s and C''_e contain 2 paths in total. It is enough to show that for every $k = 2, \ldots, s$, $C'_k \simeq C_k$ and for every $k = e, \ldots, r-1$, $C''_k \simeq C_k$. Assume that there is $l \in \{2, \ldots, s\}$ such that $C'_l \neq C_l$ and assume that l is the smallest such integer. Then C_l contains another component K which does not intersect projection of $C_1 = C'_l$. Note that we can apply Lemma 6 on K as well, i.e., there will be a level $l' > l + 1$ such that all components of $C_{l'}$ intersecting K are paths. Since, each such path will create 2 occurrences of H0H, there is only one such path P. Note that there is no other occurrence of H0H in F. It is easy to see that for all $s < k < e$, $C'_k \simeq C_s$, as any change would introduce a new occurrence of H0H. Similarly, for any $l' < k < e$, there is only one component of C_k intersecting K, P. Now, the layer C_{e-1} contains two paths and C_e only one path. Thus, the change from C_{e-1} to C_e introduces new occurrences of H0H, a contradiction. Hence, for every $k = 2, \ldots, s$, $C'_k = C_k$ and for every $k = e, \ldots, r-1$, $C''_k = C_k$. This implies that $\bar{s} = s$ and $\bar{e} = e$. The lemma follows.

Second, assume that C'_s and C''_e contain 3 paths in total. Without loss of generality assume that C'_s contains 2 paths and C''_e has only 1 path. This will create 6 occurrences of H0H in F. Therefore, as before, C_{e-1} contains two paths and C_e only one path, a contradiction. □

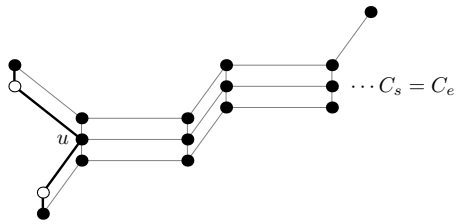


Figure 5.11: A complex component: the case when layers C_s and C_e are identical.

Observation 7. *Let F be a saturated fold of a sparse protein with no occurrences of the substrings* H0HH *and* H0H0H*, and at most six occurrences of the substring* H0H*. Let C be a complex component without a* vv*-vertex. Let $s > 2$ ($e < r - 1$) be the smallest (largest) integer such that C_s (C_e) is different from C_1 (C_r). Then $s \neq e$, i.e, the middle part of a complex component without a* vv*-vertex (layers $C_s, \ldots, C_e$) contains at least 4* S*-vertices.*

Proof. First, notice that if $s = e$ then the end point of the path u in $C_s \simeq C_e$ belongs to two different occurrences of H0H. If these two occurrences share a 0-vertex v then v connects to three vertices, a contradiction. Otherwise, we have an occurrence of substring H0H0H, cf. Figure 5.11, again a contradiction. □

Basic complex component

Definition 4 (basic complex component)**.** Let F be a saturated fold of a sparse protein with no H0HH as a substring. Let C be a complex component of F without a vv-vertex with layers $C_1, \ldots, C_r$. Let s be the smallest integer such that C_s is different from C_1 and let e be the largest integer such that C_e is different from C_r. If C_s is a path and for any $i \in s + 1, \ldots, e$, C_i is identical to C_s then we call C a *basic complex component*.

Note that a basic complex component consists of three parts stacked vertically on each other: (1) a tube or 2-layer component; (2) a wall; and (3) a tube or 2-layer component.

Observation 8. *Let F be a saturated fold of a sparse protein with no* H0HH *as a substring. Any basic complex component of F contains at least 20* S*-vertices (the lower and upper part at least 8 each and the wall at least 4) and at least 4 occurrences of substring* H0H*.*

Appendix components

In this subsection, we show that if a complex component C without vv-vertices is not basic, then its layers change exactly four times, i.e., it consists of five parts stacked on top of each

other: (1) a 2-layer component or a tube; (2) a wall; (3) a pseudo 2-layer component with exactly one vertex with plane degree 1 in each of two layers; (4) another wall; and (5) a 2-layer component or a tube. The part in the middle (3) will be called an *appendix*, and such a complex component will be called an *appendix component*. An example of an appendix component is in Figure 5.12(a). Let us start with the formal definition of an appendix component.

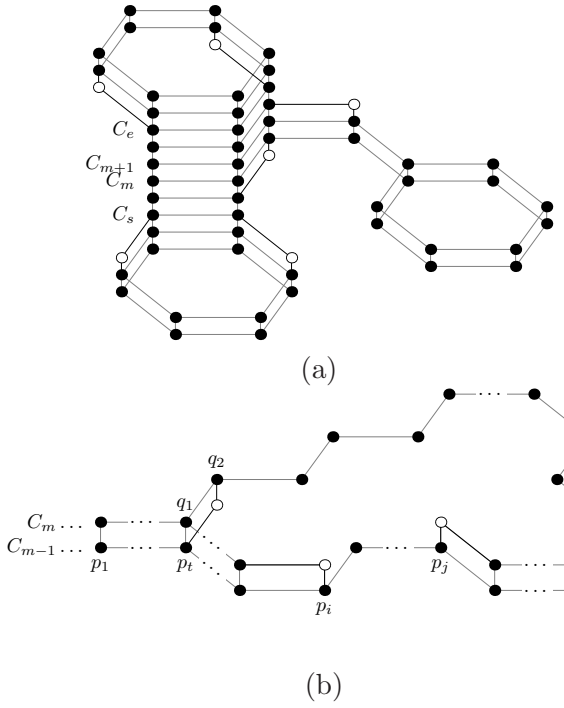


Figure 5.12: (a) An example of an appendix component and the six occurrences of H0H contained in it. (b) Illustration what happens if C_{m-1} is not a subset of C_m.

Definition 5 (appendix component)**.** Let F be a saturated fold of a sparse protein with no occurrence of the substring H0HH. Let C be a complex component of F without a vv-vertex with layers $C_1, \ldots, C_r$. Let s be the smallest integer such that C_s is different from C_1 and let e be the largest integer such that C_e is different from C_r. Assume that both C_s and C_e contain only one path, and that there is an integer $s < m < e - 1$ such that $C_s \simeq C_{s+1} \simeq \cdots \simeq C_{m-1}$, $C_m \simeq C_{m+1}$, $C_{m+2} \simeq C_{m+3} \simeq \cdots \simeq C_s$, and C_s and C_e are subsets of C_m. Furthermore, assume that C_m has exactly one vertex with plane degree 1 and this vertex is an end point of the paths in C_s and C_e. Such a complex component will be called an *appendix component* and the layers C_m and C_{m+1} we be called an *appendix*.

Consider a path in C_m (C_{m+1}) starting at the vertex with plane degree 1 and ending before the first vertex with plane degree 3. These paths in C_m and C_{m+1} will be called the *arm* of the appendix.

Note that an appendix without its arm is a proper 2-layer component or a tube with two layers.

Lemma 8. *Let F be a saturated fold of a sparse protein with no occurrence of the substring* H0HH*, and at most six occurrences of the substring* H0H*. Every non-basic complex component without a* vv*-vertex in F is an appendix component.*

Proof. Consider a complex component C in F without vv-vertices with layers $C_1, \ldots, C_r$. Assume that C is not a basic complex component. Let s (e) be the smallest (largest) integer such that C_s (C_e) is different from C_1 (C_r). By Lemma 7, both C_s and C_e contain only one path. Let m be the smallest integer such that $s < m < e$ and C_m is different from C_s.

First, we will prove that C_s is a subset of C_m. Since, $C_s \simeq C_{m-1}$, C_{m-1} is a path $P = (p_1, \ldots, p_\ell)$. Clearly, p_1^1 and p_ℓ^1 are H-vertices. Assume to the contrary that C_m is not a superset of C_{m-1}. Let p_i (p_j) be the first (last) vertex on path P such that p_i^1 (p_j^1) is a 0-vertex. Clearly, $i \neq j$, hence, we have two new occurrences of H0H. There are no other occurrences of H0H. Therefore, $C_m \simeq C_{m+1} \simeq \cdots \simeq C_e$, i.e., C_m is a path. Thus, there is a path in C_m connecting paths $(p_1^1, \ldots, p_{i-1}^1)$ and $(p_{j+1}^1, \ldots, p_\ell^1)$. Let $(q_1, \ldots, q_{\ell'})$ be a shortest such path. Then $q_1 = p_t^1$ for some $t \in \{1, \ldots, i-1\}$ and q_2^{-1} does not lie on P, i.e., it is a 0-vertex. Then the paths p_t, q_2^{-1}, q_2 forms another occurrence of H0H, a contradiction, cf. Figure 5.12(b).

Let m' be the largest integer such that $s < m' < e$ and $C_{m'}$ is different from C_e. By symmetry, we have that $C_{m'}$ is a superset of C_e. Obviously, $m \leq m' + 1$. We will show that $m \leq m'$, i.e., that there are at least two changes between layers C_s and C_e. Assume to the contrary that $m = m' + 1$. Then $C_s \simeq C_{m-1} \simeq C_{m'} \overset{\sim}{\subset} C_m$ and $C_e \simeq C_{m'+1} \simeq C_m \overset{\sim}{\subset} C_{m'}$, i.e., $C_m \simeq C_{m'}$. However, this is a contradiction with the fact that C is not a basic complex component, since we have $C_s \simeq \cdots \simeq C_{m-1} \simeq C_{m'} \simeq C_m \simeq C_{m'+1} \simeq \cdots \simeq C_e$.

Since there are at least two changes from layer C_s to layer C_e and each change will introduce at least one new occurrence of H0H, each of the two changes can create only one occurrence of H0H and there are no other changes. Therefore, there is exactly one vertex z in C_m which is a horizontal neighbor of some p_i^1 such that z^{-1} is a 0-vertex. If $i \neq 1, \ell$ then we get an occurrence of H0HH. Hence, C_m extends the copy of path P in the plane $H(C_m)$

at one of its ends. Similarly, $C_{m'}$ extends a copy of the path in layer $C_{m'+1}$ at one of its ends. Furthermore, since there are no other changes $C_m \simeq C_{m+1} \simeq \cdots \simeq C_{m'}$.

It remains to show that $m' = m + 1$ and that C_m has exactly one vertex with plane degree 1. The extended part of C_m ($C_{m'}$) does not have a vertex of plane degree one because otherwise it will be an H-vertex with three 0-neighbors. The number of vertices with odd plane degree in C_m ($C_{m'}$) is even. Since, there is only one vertex with plane degree one in C_m ($C_{m'}$), there is an odd number of vertices with plane degree 3, which implies there is at least one such a vertex, say $w \in C_m$. Now, if $m' > m + 1$ then $w^1 \in C_{m+1}$ has five H-neighbors, a contradiction. Second, if $m' = m$ then z is a vv-vertex, a contradiction. Hence, $m' = m+1$, i.e, the complex component C has a pseudo 2-layer component between two walls. It follows that C is an appendix component. □

The following observation follows by a careful examination of Figure 5.12(a).

Observation 9. *Let F be a saturated fold of a sparse protein with no occurrences of the substrings* H0HH*. Let C be an appendix component and C_s, C_m and C_e be the layers after the first, after the second and before the last change, respectively. Then $m \geq s + 2$ and $e \geq m + 3$. Each wall (layers $C_s, \ldots, C_{m-1}$ and $C_{m+2}, \ldots, C_s$) contains at least 4, the arm of appendix of C at least 4 and the appendix without arm at least 10* S*-vertices. Thus layers $C_s, \ldots, C_e$ contain at least* 22 S*-vertices.*

5.4.4 Counting in one plane

Consider a set S of vertices in a hexagonal plane. Set S naturally induces a graph in the plane in which any two neighboring vertices are connected by an edge. In the following S will represent both the set of vertices and the graph induced by this set. Assume that each vertex of S has a degree at least 2. We say that S is complete if every vertex which lies inside the boundary of S, denoted as $B(S)$, is in S as well. Let $K_{\hexagon}(S)$ be the number of hexagons which lie inside the boundary $B(S)$, $K_2(S)$ the number of vertices of degree 2 of S and $K_3(S)$ the number of vertices of degree 3. Our goal is to lower bound $K_3(S)$ by some function of $K_2(S)$. We will do that in two steps.

Lemma 9. *Let S be any set of vertices in a hexagonal plane such that each vertex of S has a degree at least 2. We have $K_3(S) \leq 2K_{\hexagon}(S) - 2c$, where c is the number of connected components of S.*

Proof. First, assume that S is a complete 2-connected set. We proceed by induction on $K_{\hexagon}(S)$. If $K_{\hexagon}(S) = 1$ then the lemma trivially holds. There must be a hexagon H in S sharing at least two sides with the boundary $B(S)$ such that all its boundary sides form a single path P. Consider a set S' obtained from S by removing inner vertices of path P. Set S' contains all hexagons contained in S besides H. Thus S' is a complete 2-connected set and the number of hexagons $K_{\hexagon}(S')$ is $K_{\hexagon}(S) - 1$. At the same time, S' must have two vertices of degree 3 less than S (end points of P become vertices of degree 2 and other vertices on P which were removed when constructing S' must have had degree 2). By induction hypothesis, $K_3(S) - 2 = K_3(S') \leq 2K_{\hexagon}(S') - 2 = 2(K_{\hexagon}(S) - 1) - 2$. This implies that $K_3(S) \leq 2K_{\hexagon}(S) - 2$.

Second, assume that S is just a 2-connected set. Let $\bar{S}$ be a set constructed from S by adding all vertices which lies inside the boundary $B(S)$. Note that $B(\bar{S}) = B(S)$ and $\bar{S}$ is complete. Furthermore, the number of vertices of degree 3 of $\bar{S}$ could only increase when adding vertices to S. Therefore, $K_3(S) \leq K_3(\bar{S}) \leq 2K_{\hexagon}(\bar{S}) - 2 = 2K_{\hexagon}(S) - 2$.

Third, assume that S is connected and let $S_1, \ldots, S_l$ be 2-connected components of S. Contracting every 2-connected component to a single vertex we obtain a tree T. Every vertex of T of degree 1 or higher than 3 must be a contracted vertex and the number of contracted vertices is l. Let n_d be the number of all vertices of degree d and let n'_d the number of all contracted vertices of degree d. Note that for $d = 1$ and $d \geq 4$, $n'_d = n_d$ and that $\sum_{d \geq 1} n'_d = l$. Set S has three types of vertices of degree 3: (i) vertices of degree 3 from 2-connected components; (ii) vertices of degree 3 created by edges attached to 2-connected components; and (iii) $n_3 - n'_3$ of vertices of degree 3 which are not part of any 2-connected component. Note that a contracted vertex of degree d in T corresponds to d vertices of degree 3 of type (ii). Therefore,

$$K_3(S) = \sum_{i=1}^{l} K_3(S_i) + \sum_{d \geq 1} d \cdot n'_d + n_3 - n'_3 = \sum_{i=1}^{l} K_3(S_i) + 2l + \sum_{d} (d-2) n_d \,.$$

It can be easily shown by induction that for any tree, $\sum_d (d-2)n_d = -2$. We know that the lemma holds for every 2-connected component, i.e., for every $i = 1, \ldots, l$, $K_3(S_i) \leq 2K_{\hexagon}(S_i) - 2$. Plugging these two facts into formula for K_3 we obtain

$$K_3(S) \leq 2 \sum_{i=1}^{l} K_{\hexagon}(S_i) - 2l + 2l - 2 = 2K_{\hexagon}(S) - 2 \,.$$

Finally, by summing the bound for each connected component of S, we obtain the desired bound for any S. □

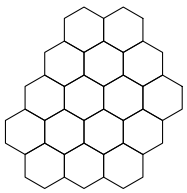


Figure 5.13: Example of a pseudohexagonal shape with sides 3,3,3,2,4,2.

Lemma 10. *Let S be any set of vertices in a hexagonal plane such that each vertex of S has a degree at least 2. We have $K_{\hexagon}(S) \leq \frac{1}{12}(K_2(S)^2 + K_2(S) - 30)$.*[1]

Proof. First, assume that S is complete and 2-connected, and that its boundary does not have two consecutive concave angles, i.e., the boundary forms a pseudohexagonal shape, cf. Figure 5.13. We will show that lemma holds for any such pseudohexagonal shape by induction on $K_2(S)$, which is now equal to the sum of its six sides (measured in the number of hexagons on the particular side). It is easy to verify that the lemma holds in the base case when there are two neighboring sides equal to one. Indeed, in this case hexagonal shape is formed by a linear chain of t hexagons and the number of vertices of degree 2 is $2t + 4$. Assume it is not a base case and let s be the shortest side of the hexagonal shape S. Observe that the neighboring sides to s are longer than 1. Consider a hexagonal shape S' obtained from S by removing a row of hexagons on the side s. The number of hexagons $K_{\hexagon}(S')$ is $K_{\hexagon}(S) - s$ and since side s was prolonged by 1, while the neighboring sides shortened by 1, $K_2(S') = K_2(S) - 1$. By induction hypothesis, $K_{\hexagon}(S) - s = K_{\hexagon}(S') \leq \frac{1}{12}(K_2(S')(K_2(S') + 1) - 30) = \frac{1}{12}(K_2(S)(K_2(S) - 1) - 30)$. Since, s is the shortest side of S, $K_2(S) \geq 6s$, and hence

$$\begin{aligned} K_{\hexagon}(S) &\leq s + \tfrac{1}{12}(K_2(S)(K_2(S) - 1) - 30) \\ &\leq \tfrac{1}{6}K_2(S) + \tfrac{1}{12}(K_2(S)^2 - K_2(S) - 30) = \tfrac{1}{12}(K_2(S)^2 + K_2(S) - 30)\,. \end{aligned}$$

Second, assume that S is complete and 2-connected. We will transform S to a new set S' by repeating the following process until possible: if there are two or three consecutive

[1] *Note that this is not a tight bound. We conjecture that the following bound holds $K_{\hexagon}(S) \leq \frac{1}{12}(K_2(S)^2 - 6K_2(S) + 12)$.*

concave angles on the boundary add the vertices of the hexagon they are part of, to S. It is easy to see that this process must stop (we will never go outside of any hexagonal shape enclosing S). Note that in each step $K_{\hexagon}$ increases by 1 and K_2 either stays the same or decreases by 1. Thus $K_{\hexagon}(S) \leq K_{\hexagon}(S')$ and $K_2(S') \leq K_2(S)$. Since, S' is a hexagonal shape and complete, the lemma holds for it. Thus it holds for S as well: $K_{\hexagon}(S) \leq K(S') \leq \frac{1}{12}(K_2(S')^2 + K_2(S') - 30) \leq \frac{1}{12}(K_2(S)^2 + K_2(S) - 30)$.

Third, assume that S is 2-connected, but not complete. Let $\bar{S}$ be the completion of S as in the proof of Lemma 9. Note all vertices of degree 2 in $\bar{S}$ are on the boundary $B(\bar{S}) = B(S)$ and they must be vertices of degree 2 in S as well. Hence, $K_{\hexagon}(S) = K_{\hexagon}(\bar{S})$ and $K_2(S) \geq K_2(\bar{S})$. Since $\bar{S}$ is complete and 2-connected, it satisfies the lemma. It follows that S satisfies the lemma as well.

Finally, we prove that any set S satisfies the lemma by induction on the number of 2-connected components. Let S' be a 2-connected component of S with at most one edge to $S - S'$. Clearly, such a component exists. If S' is not connected to $S - S'$, let $S'' = S - S'$. Otherwise, let $P = (x, \ldots, y)$ be the path such that x is the only vertex of P in S', all inner vertices $I(P)$ of P have degree 2 and y has degree 3. Then let $S'' = S - S' - I(P)$. Note that $K_{\hexagon}(S) = K_{\hexagon}(S') + K_{\hexagon}(S'')$ and $K_2(S) \geq K_2(S') + K_2(S'') - 2$. Furthermore, S'' satisfies the lemma by induction hypothesis and S' as well, since it is a 2-connected set. Easy calculations and the fact that $K_2(S'), K_2(S'') \geq 6$ show that S satisfies the lemma as well. □

Corollary 1. *Let S be any set of vertices in a hexagonal plane such that each vertex of S has a degree at least 2. We have $K_3(S) \leq \frac{1}{6}(K_2(S)^2 + K_2(S) - 30) - 2c$, where c is the number of connected components of S.*

5.4.5 Limiting certain types of connections and vertices

In this subsection we limit certain types of connections and vertices that occur in a saturated fold F of q. We first prove that there are at most 4 v-vertices in F.

Claim 6. *Let F be a saturated fold of q and assume it contains a complex component C without a vv-vertex. Let s be the smallest integer such that C_s is different from C_1 and let e be the largest integer such that C_e is different from C_r. Let w_1 be the length of the path in C_s and w_2 the length of the path in C_e. Then $w_1 + w_2 \leq 40$.*

Proof. First, note that w_1 and w_2 are well-defined, as by Lemma 8, C_s and C_e contain only one path. Let $(p_1, \ldots, p_{w_1})$ be the path in C_s. Obviously, vertices $p_1^{-1}, \ldots, p_{w_1}^{-1}$ are h-vertices. Let p_0^{-1} ($p_{w_1+1}^{-1}$) be the other neighbor of p_1^{-1} ($p_{w_1}^{-1}$). Both, p_0^{-1} and $p_{w_1+1}^{-1}$, are vh-vertices, otherwise we have an occurrence of substring H0HH. Similarly, all vertices, $p_0^{-s+1}, p_1^{-s+1}, \ldots, p_{w_1+1}^{-s+1}$, are vh-vertices. Therefore, in layers C_1 and C_s we have at least $w_1 + 4$ S-vertices. Similarly, in layers C_e and C_r we have at least $w_2 + 4$ S-vertices. Hence, by Observation 7, C contains at least $w_1 + w_2 + 12$ S-vertices. Since q contains 52 S-vertices , the claim follows. □

Lemma 11. *Let F be a saturated fold of q. No* v*-vertex can be part of substring* $(\mathsf{00HH})^{356}$*. Consequently, there are at most 4* v*-vertices in F.*

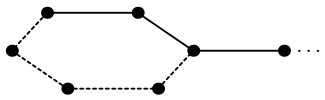


Figure 5.14: An example of extending the wall's end in layer eliminating vertices with horizontal degree 1.

Proof. Note that since each complex component introduces at least 4 occurrences of H0H, there is at most one complex component in F. Assume to the contrary that the substring $(\mathsf{00HH})^{356}$ contains a v-vertex. By Lemma 4, the substring contains only v-vertices. Let $P_1, \ldots, P_k$ be all hexagonal planes containing these v-vertices and let S_i be the set of H-components in the plane P_i which contain at least one of these v-vertices and let S be the union of $S_1, \ldots, S_k$. Since every component is either a tube, a 2-layer component, a complex component with six vv-vertices, a basic complex component or an appendix complex component, we have the following observations:

- The set S contains only layers of 2-layer components, complex components with vv-vertices, the lower and upper parts of a complex components without vv-vertices if they are 2-layer components and layers of appendix of appendix components. Since all these layers come in identical pairs with exception of a vv-component in which 2-layers differ in 6 vertices, we will consider only one layer in the pair. From each pair select only one layer, for the vv-component select the layer with vv-vertices. Let $J \subseteq \{1, \ldots, k\}$ be the set of the selected layers and let $M = \cup_{i \in J} S_i$. We have $K_2(M) \leq K_2(S)$ and $K_3(M) \geq \frac{1}{2} K_3(S)$.

- All vertices have horizontal degree 2 or 3 with exception of the wall and (possibly) appendix of a complex component without vv-vertices. The layer of a wall without appendix contains two vertices with horizontal degree 1, but no vertex with horizontal degree 3, hence, it is not included in M. On the other hand, a layer containing the appendix contains exactly one vertex with horizontal degree 1. Let us extend the path ending in this vertex in its layer until we join another H-vertex, see an example in Figure 5.14. There is always a way to do this which introduces at most 4 new vertices with horizontal degree 2, and eliminates at least one such vertex. Let M' be the set M extended by these elements and S'_i either S_i or S_i extended by these elements if S_i was the component containing the appendix. Hence, since there is at most one complex component and it contains at most two layers with appendix, we have $K_2(M') \leq K_2(M) + 3$ and $K_3(M') \geq K_3(M)$.

By Corollary 1, we have

$$\begin{aligned} K_3(M) \leq K_3(M') = \sum_{i \in J} K_3(S'_i) &\leq \frac{1}{6} \sum_{i \in J} (K_2(S'_i)^2 + K_2(S'_i)) - 7k \\ &\leq \frac{1}{6}(K_2(M')^2 + K_2(M')) - 7 \leq \frac{1}{6}(K_2(M)^2 + 7K_2(M)) - 5\,. \end{aligned} \tag{5.1}$$

It remains to upper bound the number of vertices with horizontal degree 2. Such vertices are either vh-vertices or h-vertices. By Observation 2, there is at most 52 vh-vertices. If we examine all possible components, we can see that h-vertices are in the inner layers of tubes or in the last (first) layer of the lower (upper) part of the complex components which are directly attached to the walls. However, the component in the inner layer of tube contains only vertices with horizontal degree 2, hence, it does not belong to S. Since we have at most one complex component, by Claim 6, we have at most 40 h-vertices which are in S. At most half of these vertices are in M, hence, $K_2(M) \leq (52 + 40)/2 = 46$. By (5.1), we have

$$K_3(S) \leq 2K_3(M) \leq \frac{1}{3}(46^2 + 7 \times 46) - 10 < 711\,.$$

Since, every v-vertex has horizontal degree 3, by the assumption, we have $K_3(S) \geq 2 \times 356 = 712$, a contradiction. □

(S ≍ h)-connections

Corollary 2. *Let F be a saturated fold of q. Then F contains* 36 *(S ≍ h)-connections.*

Proof. By Observation 2, F contains 36 ($\mathsf{S} \asymp \mathsf{D}$)-connections. Each D-vertex in such a connection is part of the substring $(\mathsf{00HH})^{356}$, hence, by Lemma 11, is an h-vertex. □

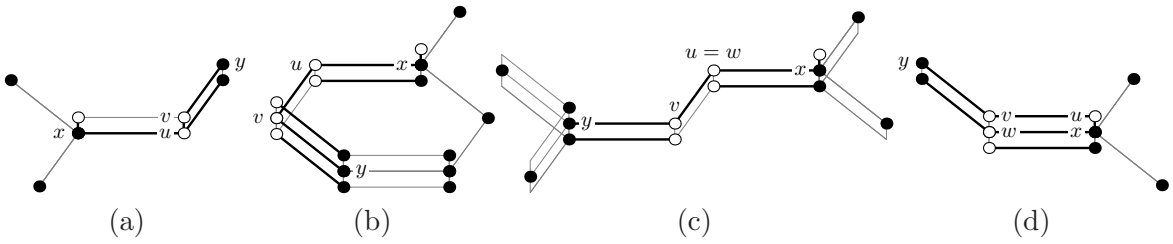


Figure 5.15: (a-c) Illustration of an external horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection. Contradictory cases: (a) the case when $v = u^1$, (b) the case where x and y are on the same hexagon. The only possible configuration in (c). (d) Illustration of a vertical external ($\mathsf{S} \asymp \mathsf{h}$)-connection.

We define two types of ($\mathsf{S} \asymp \mathsf{h}$)-connections. Assume that S-vertex x is ($\mathsf{S} \asymp \mathsf{h}$)-connectedand to y. We say that this ($\mathsf{S} \asymp \mathsf{h}$)-connection is *horizontal* is x and y are on the same plane (cf. Figure 5.15(c)) and it is *vertical* if x and y are on two consecutive planes (cf. Figure 5.15(d)).

Claim 7. *Let F be a saturated fold of q. Let x be a vh-vertex and y be an h-vertex in two different components W_1 and W_2. Then all ($\mathsf{S} \asymp \mathsf{h}$)-connections are either* horizontal *or* vertical*. Furthermore, a vertical ($\mathsf{S} \asymp \mathsf{h}$)-connection creates an $\mathsf{H0H}$-connection between x and a vertical neighbor of y. Finally, if W_1 and W_2 are non-complex, there is at most one parallel ($\mathsf{S} \asymp \mathsf{h}$)-connection with (x, u, v, y) and in the vertical case the two components share only one layer.*

Proof. Let x be on plane H_i. Without loss of generality assume that x^1 is a $\mathsf{0}$-vertex and let w be the horizontal $\mathsf{0}$-neighbor of x. Clearly, u is either x^1 or w. We consider each case separately.

Case 1 ($u = w$). If $v = u^1$ then y must be a horizontal neighbor of v and thus, u is adjacent to the H-vertex y^{-1}, a contradiction (cf. Figure 5.15(a)). Furthermore, if $v = u^{-1}$ then $y = x^{-1}$ and it follows that x and y are in the same component, a contradiction. Therefore, v is a horizontal neighbor of u and y is a horizontal neighbor of v. Note that y must be the horizontal neighbor of v that is not on the same hexagon with x otherwise, x and y would be in the same component, a contradiction, cf. Figure 5.15(b). Hence, x and y are on the same plane (horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection), cf. Figure 5.15(c). Next, assume that (x^i, u^i, v^i, y^i) and (x^j, u^j, v^j, y^j) are two parallel ($\mathsf{S} \asymp \mathsf{h}$)-connections with (x, u, v, y). Obviously, $i, j < 0$,

and let $i < j$. Since (x, u, v, y) and (x^i, u^i, v^i, y^i) are parallel connections, all vertices between x and x^i (y and y^i) are H-vertices, i.e., neither x^j nor y^j is an vh-vertex. If the components they are contained in are non-complex, they must be D-vertices, a contradiction.

Case 2 ($u = x^1$). By a similar argument used in the first case we can show that $v \neq u^1$. Therefore, v is a horizontal neighbor of u. Since y is an h-vertex, none of its vertical neighbors can be a 0-vertices hence, $v = w^1$. It follows that y is a horizontal neighbor of v and it is on plane H_{i+1}, cf. Figure 5.15(d). This type of (S $\asymp$ h)-connection is called a vertical (S $\asymp$ h)-connection. Furthermore, in this setting (y^{-1}, w, x) form an H0H-connection. Second, note that y^{-1} is an S-vertex. If the component containing y^{-1} is non-complex, then it is a vh-vertex, i.e., y^{-2} is 0-vertex and the two components can share only one layer. Consequently, there is at most one parallel (S $\asymp$ h)-connection to (x, u, v, y). □

H0H-connections

Definition 6. We say that an H0H-connection is horizontal, vertical if both peptide edges of the connection are horizontal, vertical, respectively.

We have the following simple observation.

Observation 10. *Let F be a saturated fold of a sparse protein of length at least 5. Then every* H0H*-connection connecting two different components is either horizontal or vertical.*

Proof. Assume that H0H-connection (x, y, z) is neither horizontal nor vertical. Without loss of generality, assume that the edge (x, y) is vertical, let $y = x^1$, and (y, z) is horizontal. If z^{-1} is a 0-vertex then we have a closed path of length 4. If z^{-1} is an H-vertex then x and y belong to the same component. □

Claim 8. *Let F be a saturated fold of q and let C be a component of F. Assume that C_i is a layer in F that does not contain any vertex of plane degree 1. Then there is no* H0H*-connection with both end points in C_i. Consequently, there is no internal* H0H*-connection in a tube or a 2-layer component.*

Proof. To the contrary assume that x and y have a common horizontal 0-neighbor z. We remark that component C cannot be a vv-component since such a component already contains 6 H0H-connections which are different type than (x, z, y). Clearly one of the vertical

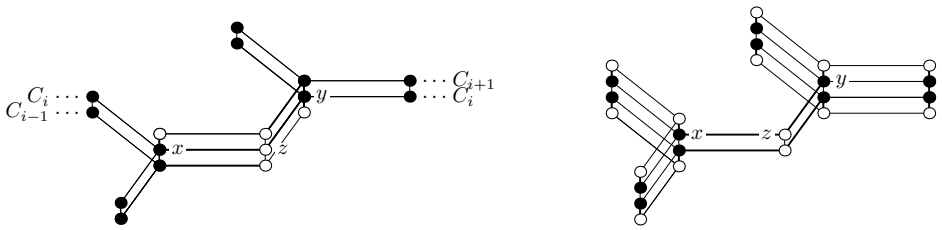


Figure 5.16: Horizontal H0H-connection (x, z, y): (a) the case where y^{-1} is 0-vertex, (b) the case where y^1 is 0-vertex.

neighbors of x has to be a 0-vertex otherwise F contains an occurrence of H0HH as a substring. Without loss of generality assume that x^1 is a 0-vertex. Similarly one of the vertical neighbors of y has to be a 0-vertex. First assume that y^{-1} is a 0-vertex, cf. Figure 5.16(a). Note that in this case, layers C_{i-1}, C_i and C_{i+1} are different which cannot happen in any component of F. Therefore, x and y are in different components, a contradiction.

Second assume that y^1 is a 0-vertex. It follows that y^{-1} is an H-vertex. Note that x^{-2} and y^{-2} are 0-vertices, otherwise F would contain H0HH as a substring cf. Figure 5.16(b). Moreover, all horizontal neighbors of y^1, y^{-2}, x^1 and x^{-2}, except z^1 and z^{-1} are 0-vertices, otherwise F would contain an occurrence of the substring H0H0H. Next consider the H0H connection $(x,\ z,\ y)$. One of the vertices x and y has to connect to a D-vertex w via two 0-vertices u and v. By Lemma 11, w must be an h-vertex. It is easy to see that $u = x^1$ and $v = x^2$. Now w must be a horizontal neighbor of x^2 which is not possible. □

Corollary 3. *Let F be a saturated fold of q. Then the smallest non-simple tube contains* 7 *hexagons and* 36 S*-vertices, cf. Figure 5.8(b).*

Lemma 12. *Let F be a saturated fold of q. Consider an* H0H*-connection (x, y, z) connecting two non-complex components W_1 and W_2. If this connection is horizontal then at least one of the two components is a tube with more than two layers, they share only one plane and they are configured as in Figure 5.17(b). If this connection is vertical then they do not share any plane.*

Proof. First, assume that (x, y, z) is a horizontal H0H-connection. It is easy to see that W_1 and W_2 make another horizontal H0H-connection (x', y', z'), cf. Figure 5.17(a). By the properties of q one of the vertices x or z must connect to a D-vertex w through two 0-vertices

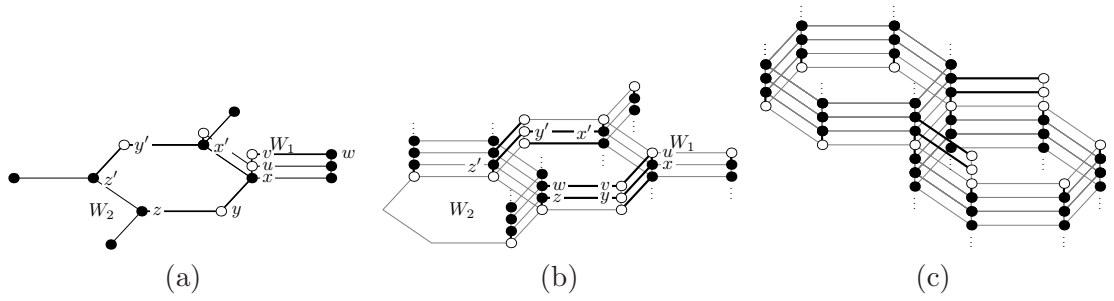


Figure 5.17: Situation when two non-complex components are connected with a horizontal H0H-connection: (a) x is connected to an h-vertex w away from the other component; (b) w belongs to the other component. (c) An example of two non-complex components connected with a vertical H0H-connection.

u and v. Without loss of generality, let it be x. Obviously, x is a vh-vertex. Without loss of generality, assume that $u = x^1$. By Lemma 11, w must be an h-vertex, therefore, w is a horizontal neighbor of v. Now, if $v = u^1$ then u will be adjacent to the H-vertex w^{-1} (cf. Figure 5.17(a)), a contradiction. Hence, v is a horizontal neighbor of u and it is easy to see that $v = y^1$ and $w = z^1$. The configuration of parts of two components is depicted in Figure 5.17(b). Since, the h-vertex w belongs to W_2, W_2 must be a tube with height more than 2 layers and since these two components are non-complex, they can only share one plane.

Second, assume that (x, y, z) is a vertical H0H-connection. Obviously, the two components do not share any plane, and all H0H-connections between them are vertical. An example of configuration in which two non-complex component are vertically H0H-connected is depicted in Figure 5.17(c). □

5.4.6 Limiting the possible configurations of complex components

In this subsection we show that only a limited number of configurations are possible for a complex component. This will greatly simplify our analysis in the later sections. In the following arguments we say that a path has length k if it contains k vertices.

Lemma 13. *Let F be a saturated fold of q. Then F does not contain any* vv*-component.*

Proof. Let C be a vv-component. Consider any of the H0H-paths in C for example (x_1, x_1^1, y_1^1), cf. Figure 5.9. Notice that this path has to continue with substring $(00HH)^{k_i}$ at one end. By Lemma 11, all H-vertices in this substring are h-vertices, i.e., either y_1^1 or x_1^{-1} has to

00-connect to an h-vertex. It is easy to check that none of these connections is possible, a contradiction. □

Lemma 14. *Let F be a saturated fold of q and let C be a complex component in F with layers $C_1, C_2, \ldots, C_r$. Layer C_1 and similarly C_r is either one hexagon or consists of two hexagons attached by one edge or connected by a path (cf. Figure 5.19).*

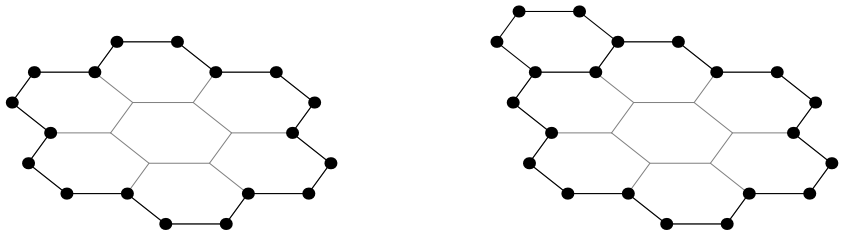


Figure 5.18: (a) The second smallest cycle without H0H occurrences. (b) The smallest possible layer C_1 of a complex component with the lower part being a 2-layer component containing a large cycle.

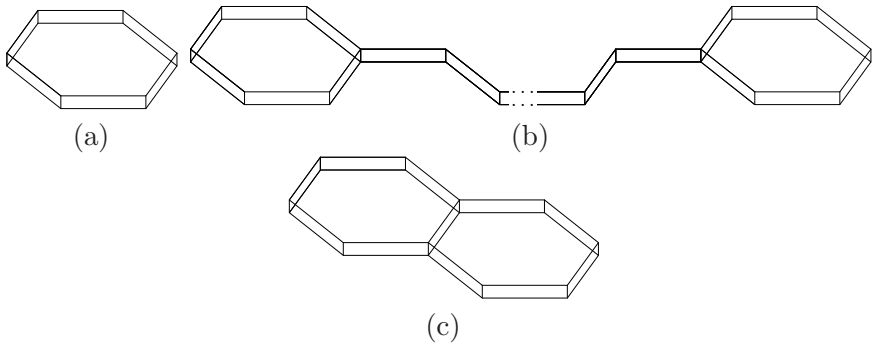


Figure 5.19: Possible configurations for the upper and lower part of a complex component.

Proof. By Lemma 13, C does not contain any vv-vertex. We prove the claim for C_r, the proof for C_1 follows by symmetry. By Lemma 8, C_r does not contain any horizontal H0H-connection. Furthermore, C_r cannot contain more than 2 vertices of plane degree 3 because otherwise we get more than 4 v-vertices, a contradiction by Lemma 11. Hence, C_r has one of the three topologies depicted in Figure 5.19. In principle, each hexagon could be replaced with larger cycle; we will show that this does not happen.

The smallest possible component layer with no vertex of plane degree 3 other than a simple hexagon is a cycle containing 7 hexagons inside, cf. Figure 5.18(a) and the smallest

possible layer with exactly two vertices of plane degree 3 and at least three hexagons is depicted in Figure 5.18(b). We prove that C_r cannot be the cycle in Figure 5.18(a) by computing the lower bound on the number of S-vertices in F. Clearly C will have more S-vertices if C_r has two vertices of degree 3 as in Figure 5.18(b).

Assume the contrary. We will consider two cases: C is either a basic or an appendix complex component.

Case 1. Let C be a basic complex component. Note that the number of S-vertices in C is minimized when the wall width is maximized and the wall height is minimized. The lower part of C is either a simple tube or the second smallest tube similar to C_r. Figure 5.20(a)-(b) depicts these configurations with the smallest number of S-vertices. The width of the wall can be at most 4 and 16 in the first and second configurations, respectively. However, in both of these configurations the number of S-vertices is at least 44 which happens when the height of the wall is 2. In addition, notice that C only contains 4 H0H-connections, therefore, F must contain another component which brings the total number of S-vertices up to at least $44 + 12 = 56$, a contradiction by Observation 2.

Case 2. Let C be an appendix component and let w_1 and w_2 be the lower and the upper wall width of C, respectively cf. Figure 5.20(c). Similar to case 1, the lower part of C is either a simple tube or the second smallest tube. If it is the second smallest tube then the minimum number of S-vertices will be $(18 + 2) \cdot 2$ (vertices in lower and upper part) + 22 (vertices in appendix and wall ends) = 62, a contradiction. Hence, assume that the C_1 consists of one hexagon. Note that $w_1 \leq 4$ and $w_2 \leq 16$. The minimum number of S-vertices in different layers of C is as follows:

- *vertices in C_r:* 18
- *vertices in the first layer of upper part:* $18 - w_2$
- *vertices on wall ends:* 8
- *vertices of the appendix:* the appendix without the arm contains at least 10 S-vertices, the arm on its ends contain 4 and if the walls have different widths, then on the side of the shorter wall the arm has additional $|w_2 - w_1|$ S-vertices. Hence, in total appendix has at least $14 + |w_2 - w_1|$ S-vertices.
- *vertices of the first layer of the upper part:* $6 - w_1$

- *vertices in C_1:* 6

Hence, the total number of S-vertices is at least $70-w_1-w_2+|w_2-w_1|$. Now, if $w_1 \leq w_2$ then the minimum number of S-vertices is $70 - w_1 - w_2 + |w_2 - w_1| = 70 - 2w_1 \geq 62$, a contradiction. If $4 \geq w_1 > w_2$ it is $70 - w_1 - w_2 + |w_2 - w_1| = 70 - 2w_2 \geq 62$, also a contradiction. □

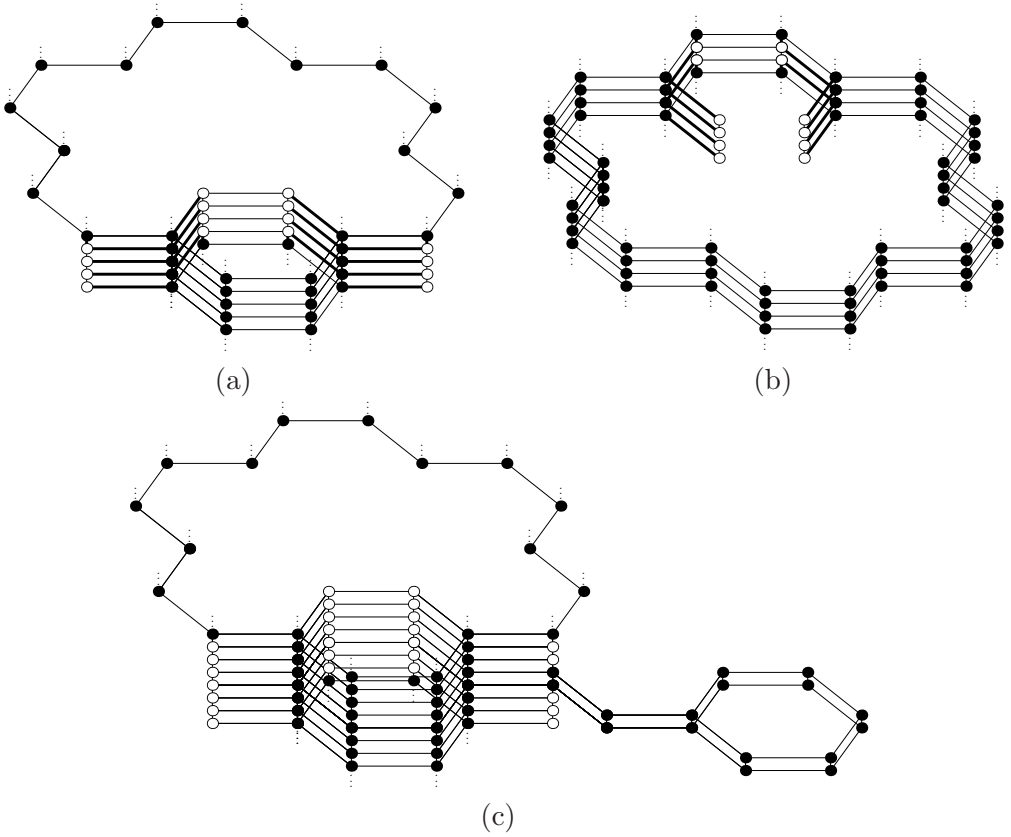


Figure 5.20: (a) A basic complex component with the second smallest tube as upper part and a simple tube as the lower part. (b) A basic complex component with the second smallest tube as upper and lower part. (c) An appendix component with the second smallest tube as upper part and a simple hexagon as lower part.

Lemma 15. *Let F be a saturated fold of q and let C be a complex component in F. Then the lower and upper part of C are simple tubes.*

Proof. By Lemma 14, the upper part of C is either a simple tube or one of the 2-layer components depicted in Figure 5.19(b)-(c). Suppose that one of the parts is not a simple

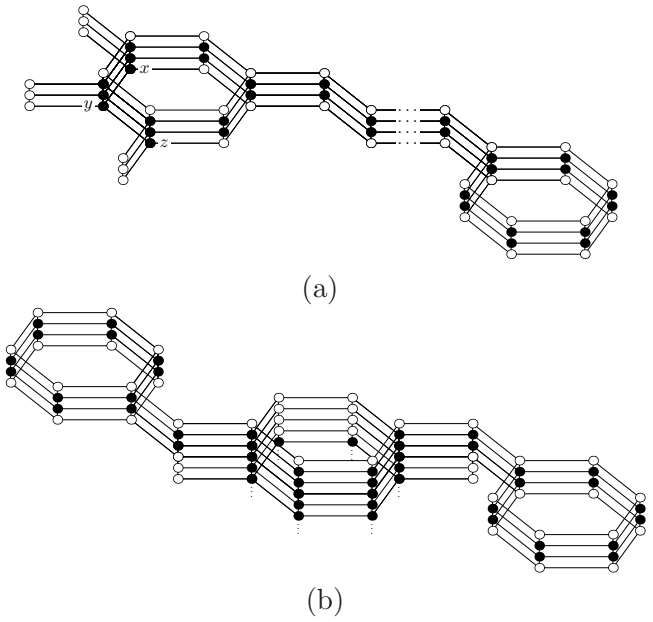
tube, say the upper part. First notice that either of the 2-layer components in Figure 5.19(b) and Figure 5.19(c) contain 4 v-vertices therefore, by Lemma 11, C cannot have an appendix, and the lower part must be a simple tube as well. Therefore, C only contains 4 H0H connections, and hence, F must contain at least one other component T. Furthermore, T has to be a simple tube because if it is a 2-layer component, a complex component or a large tube then F would contain more than 4 v-vertices, more than 6 H0H-connections or more than 52 S-vertices, respectively.

Next we consider two cases for the shape of the upper part of C:

Case 1. Assume that the upper part of C is a connector. By Lemma 6, the width of the wall is 2. Now independent of the height of the wall in C the number of D-vertices modulo 6 in F is 2, a contradiction.

Case 2. Assume that the upper part of C consists of two hexagons connected by a path, cf. Figure 5.19(c). The wall part of C can either attach to one of the hexagons or the path P connecting the two hexagons, cf. Figure 5.21. Similar to the previous case if the width of the wall is 2 the number of D-vertices modulo 6 in F is 2 independent of height or the location of the wall, a contradiction. Furthermore, if the wall is attached to one of the hexagons then by Lemma 6, the width of the wall can be at most 3. Figure 5.21(a) depicts this configuration with wall width equal to 3. Let x, y and z be the vertices on the last layer of the wall. Each of the vertices x^1 and z^1 must connect to a D-vertex via a peptide bond. The only D-vertex in their neighborhood is y^1 thus, x^1 and z^1 must both connect to y^1 which is not possible. Using a similar argument we can show that the width of the wall cannot be 3 for the case where it is attached to the path P. Since the lower part is a simple tube the only case remaining for analysis is the configuration in which the wall is attached to P and its width is 4. By Lemma 6, the smallest length of P is 6 and by Observation 7, the smallest height of the wall is 4. Note that such a component would have 40 S-vertices (28 upper part, 4 wall and 8 lower part), and with the extra component at least 52 S-vertices. Increasing either the length of the path or the height of the wall would increase this number hence, Figure 5.21(b) depicts the only possible configuration of the complex component. We show that this configuration is also impossible by determining the maximum number of (S $\asymp$ h)-connections possible. Note that at most 12 internal (S $\asymp$ h)-connections are possible across the vertices of C and T, respectively. Therefore, by Corollary 2 we need to create at least 12 external (S $\asymp$ h)-connections between the S-vertices of C and h-vertices of T. However, at least 10 of these (S $\asymp$ h)-connections must be horizontal because each vertical

($\mathsf{S} \asymp \mathsf{h}$)-connection create an $\mathsf{H0H}$-connection, by Claim 7. Since a horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection between C and T is possible only when the H-vertices in the connection are on the same plane, C and T must have at least 5 connections per plane which is easy to see it is not possible given the shape of C. □



(a)

(b)

Figure 5.21: Examples of complex components with a 2-layer component consists of two hexagons connected by a path as the upper part: (a) wall is attached to one of the hexagons; (b) wall is attached to the path connecting hexagons.

Lemma 16. *Let F be a saturated fold of q and let C be a complex component in F. The width of the wall in C is either* 2 *or* 4.

Proof. By Lemma 13, C does not contain any vv-vertex and by Lemma 15, its lower and upper part are simple tubes. Assume that the lower wall starts at layer C_s of C. First observe that the wall width cannot be 1 or 5 otherwise, we get an H-vertex with three $\mathsf{0}$-neighbors or a $\mathsf{0}$-vertex with three H-neighbors, respectively, both contradictions.

Therefore, it is enough to show that the wall width cannot be 3. Let x, y, z be the path of the wall in layer C_s (attached to the tube component). Note the number of D-vertices in this layer and above is odd. Since they have to form pairs, y^{-1} has to connect to y, and hence, x and z have to connect to x^1 and z^1, respectively. Let us look at patterns of vertical connections between consecutive layers of a tube. It can be shown by induction (from the

top of the tube) that only the patterns depicted in Figure 5.22(a) are possible. However, the pattern required to realize connections xx^1 and zz^1, depicted in Figure 5.22(b) cannot be obtained, a contradiction. □

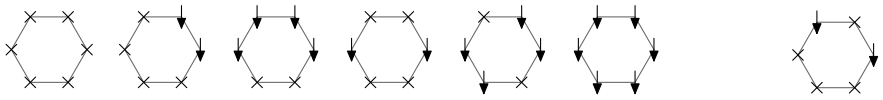


Figure 5.22: (a) All possible patterns (up to rotation) for vertical connections between two consecutive layers of a simple tube. The "x" means vertical connection is not present, arrow means it is present. (b) Pattern required to connect to the last layer of a simple tube which is connected to a path of length 3.

5.4.7 There is no appendix component

Consider an appendix component C in a saturated fold F of q. By Lemma 15, the upper and lower part of C are simple tubes. Let C_a and C_{a+1} be the layers of C that contain the appendix part. Observe that C_a and similarly C_{a+1} contains an odd number of vertices of plane degree 3 (such vertices correspond to v-vertices in C). Therefore, C contains $4k-2$ v-vertices for some positive integer k. By Lemma 11, F contains at most 4 v-vertices hence, the appendix part of C contains one hexagon.

Observation 11. *Let F be a saturated fold of q. Let C be an appendix component in F. Then C contains exactly* 2 v*-vertices.*

Lemma 17. *Let F be a saturated fold of q. Then F does not contain any appendix component.*

Proof. Assume that F contains an appendix component C. First we show that F can only contain simple tubes. By Observation 9 and Corollary 3, F cannot contain a non-simple tube, otherwise we have too many S-vertices. If F contains another complex component, then we have at least 10 H0H substrings, which is not possible. If it contains a 2-layer component, then F contains at least 6 v-vertices, two in C and 4 in the 2-layer component, a contradiction. Hence, all other components of F are simple tubes. Let N_t be the number of simple tubes.

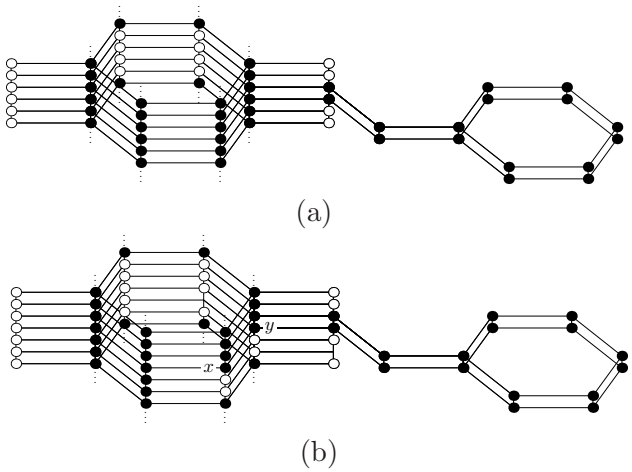


Figure 5.23: A part of (a) an appendix component with wall of width 4 all along; (b) an appendix component with wall of width 4 and 2 on different sides of appendix.

Let w_1 (h_1) be the width (height) of the lower wall of C and w_2 (h_2) the width (height) the upper wall. Let a be the lengths of the arm. We will calculate the number of D-vertices modulo 6 and the number of S-vertices in C and F. The lower (upper) part of C (a tube) contains $w_1 \bmod 6$ ($w_2 \bmod 6$) D-vertices modulo 6, the lower (upper) wall $(w_1-2)h_1$ ($(w_2-2)h_2$), the arm of appendix w_1-1+w_2-1 and the remaining part of the appendix, by Observation 11, 2 D-vertices. That is

$$2(w_1+w_2)+(w_1-2)h_1+(w_2-2)h_2 \bmod 6 \tag{5.2}$$

D-vertices modulo 6 in C, and since all other component are simple tubes, the same number in F. The number of S-vertices is $12-w_1$ ($12-w_2$) in the lower (upper) part, $2h_1$ ($2h_2$) in the lower (upper) wall, $2a-w_1-w_2+2$ in the arm and at least 10 in the remaining part of the appendix. That is at least $36+2(h_1+h_2+a-w_1-w_2)$ S-vertices in C, and $36+2(h_1+h_2+a-w_1-w_2)+12N_t$ in F.

By Lemma 16, both w_1 and w_2 are either 2 or 4, hence, we will consider the following 3 cases (without loss of generality, we assume that $w_1 \leq w_2$).

Case 1. $w_1 = w_2 = 2$. By the above formula, the number of D-vertices modulo 6 in F is 2, a contradiction with Observation 2.

In the remaining two cases, we will first show that C is the only component, i.e., that $N_t = 0$.

Case 2. $w_1 = w_2 = 4$, cf. Figure 5.23(a). By the above formula, the number of D-vertices

in F is $4+2(h_1+h_2)$ mod 6. Since, by Observation 2, this number is 4, we have $h_1+h_2 \equiv 0$ mod 3. Since, by Observation 9, $h_1, h_2 \geq 2$, we have $h_1 + h_2 \geq 6$. Also note that $a \geq 5$. Hence, the number of S-vertices is at least $36 + 2 \times 3 + 12N_t$. Since this number should be 52, we have $N_t = 0$.

Case 3. $w_1 = 2$ and $w_2 = 4$, cf. Figure 5.23(b). The number of D-vertices modulo 6 in F is $2h_2$ mod 6. Hence, $h_2 \geq 3$. And since $w_2 = 4$, we have again $a \geq 5$. Therefore, the number of S-vertices in F is at least $36 + 2 \times 5 + 12N_t$. Hence, again $N_t = 0$.

We will determine the maximum number of (S $\asymp$ h)-connections in F. Notice that in any of the configurations the S-vertices in the wall except for the end vertices on the first and the last layers cannot connect to any h-vertex, so there are at most 4 (S $\asymp$ h)-connections involving the S-vertices of the wall components. Furthermore, the S-vertices in the appendix part and its arm can only connect to the h-vertices in the wall that are in the same plane with them otherwise, we get additional H0H-connections, a contradiction. Therefore, there can be at most 4 (S $\asymp$ h)-connections involving the S-vertices of the appendix part and its arm. The last (S $\asymp$ h)-connections that we can get in F are through vh-vertices of the lower and upper tubes, which are 16 in the first configuration and 18 in the second configuration. Two more (S $\asymp$ h)-connections are possible in the second configuration through vh-vertices x and y in Figure 5.23 (b). Therefore, in total F can contain at most 28 (S $\asymp$ h)-connections, a contradiction, by Corollary 2. □

No other type of possible components can introduce 6 occurrences of H0H, hence, a saturated fold of F contains at least two components. On other hand, since any of possible components has at least 12 S-vertices, we have the following corollary.

Corollary 4. *Any saturated fold of q has at least 2 and at most 4 components.*

In what follows we will analyze all three possibilities. But first, let us have a closer look at tubes.

5.4.8 Tubes

Lemma 18. *Let F be a saturated fold of q. Any tube in F has either 12 or at least 36* S*-vertices.*

Proof. Obviously, any cycle in a hexagonal plane has at least 6 vertices, i.e., a smallest possible tube will have at least 12 S-vertices. Furthermore, by Claim 8, there is no H0H

with both ends in the same tube. The smallest cycle larger than a hexagon such that no two non-adjacent vertices are at distance two contains 7 hexagons inside. Thus, the second smallest tube has 36 S-vertices.

□

Lemma 19. *Let F be a saturated fold of q. Two* H0H*-connected tubes in F are both simple and furthermore, they make exactly two* H0H*-connections.*

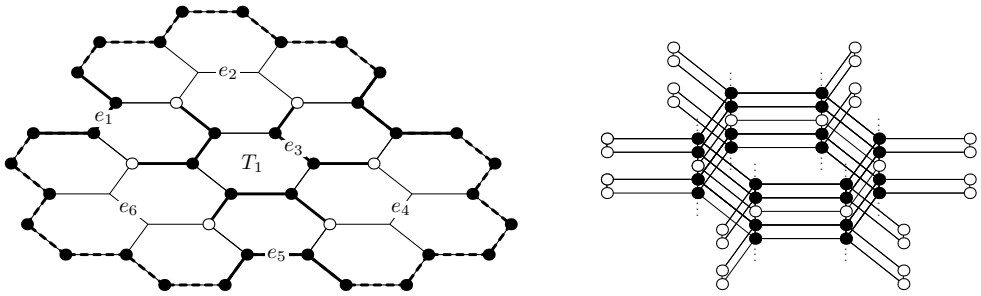


Figure 5.24: (a) A shortest possible collection of paths connecting the parts of cycle of T_2 that make 6 horizontal H0H-connections with a simple tube T_1. (b) Six vertical H0H-connections between two simple tubes.

Proof. Let T_1 and T_2 be two tubes in F. By Corollary 3, one of them, assume T_1, must be a simple tube. First note that if T_2 is not a simple tube it must make 6 H0H-connections with T_1 since F cannot have another component. Assume that there is an H0H-connection (x, y, z) such that x and z are H-vertices in T_1 and T_2, respectively. By Observation 10, there are two cases:

Horizontal H0H-connection (x, y, z)**.** By Lemma 12, T_1 and T_2 share only one plane H_i and create at least two H0H-connections as depicted in Figure 5.17(b). We will show that T_2 must also be a simple tube. Assume the contrary. Since T_2 has at least 36 S-vertices, there are no other components in F, and hence, T_2 must make 6 H0H-connections with T_1. Moreover, since T_1 and T_2 share a plane H_i no vertex of T_1 can be directly above/below any vertex of T_2, i.e, all the H0H-connections are horizontal and they are on plane H_i. Therefore, the only way how to make 6 horizontal H0H-connections is when the large cycle C_2 of T_2 on plane H_i contains 3 parts depicted in Figure 5.24(a) with thick lines. The shortest collection of 3 disjoint paths which do not create H0H-connection and connecting these parts to one

cycle is shown with dashed lines. Note that C_2 would contain at least 30 vertices and hence, T_2 would have more than 60 vh-vertices, a contradiction.

Vertical H0H-connections. Assume that x, y and z are on three consecutive planes H_i, H_{i+1} and H_{i+2}, respectively. In this case, T_1 and T_2 do not share any plane and hence, all the H0H-connections between them must be vertical and in the same planes. Note that if T_2 is not a simple tube then its projection can overlap with with the projection of T_1 on at most 3 edges creating at most 4 H0H-connections, a contradiction since there are no other components in F. Clearly, projections of two simple tubes could overlap either on 1 or 6 edges creating 2 or 6 H0H-connections, respectively. We show that two simple tubes cannot make 6 vertical H0H-connections. Assume the contrary. Figure 5.24(b) depicts two H0H-connected simple tubes with 6 H0H-connections. Note that no pair of H0H-connections in this configuration can connect through two 0-vertices. Therefore, F does not contain the substring H0H00H0H which is in q, a contradiction. □

5.4.9 2 components

Lemma 20. *Let F be a saturated fold of q. Fold F cannot have only 2 components.*

Proof. Assume there are two components in F. Six cases are possible.

Case 1. Assume they are both tubes. By Lemma 19, they can only make two H0H-connections, a contradiction.

Case 2. Assume they are both 2-layer components. By Lemma 11, we have no occurrence of substring $(00HH)^{k_i}$, a contradiction.

Case 3. Assume they are both basic complex components. Then we have 8 occurrences of H0H, a contradiction.

Case 4. Assume one component is a tube T and the other a 2-layer component C. By Lemma 11 and Lemma 18, there are only two configurations with 52 S-vertices. The first configuration consists of a connector and a tube that is the second smallest tube with 7 hexagons inside of its boundary on each layer ($16 + 36 = 52$). The second configuration consists of a 2-layer component with two simple tubes connecting by a path of length 11 and a simple tube ($40 + 12 = 52$). Note that in both configurations the two components must make 6 H0H-connections. In the first configuration it is easy to see that at most two horizontal H0H-connections can be created between the tube and connector. Therefore, all the H0H-connections must be vertical. However, in this case a v-vertex of the 2-layer

component will be part of an H0H-connection creating the substring H0HH, a contradiction. We show that the second configuration is not possible by showing that the maximum number of (S $\asymp$ h)-connections in F is less than 36. Notice that at most two h-vertices of each side of the tube can make (S $\asymp$ h)-connections with vh-vertices of the 2-layer component and hence, we can obtain at most 12 external (S $\asymp$ h)-connections between the 2-layer component and the tube. Considering the 12 internal (S $\asymp$ h)-connections in the tube, the total number of (S $\asymp$ h)-connections in this configuration is at most 24, a contradiction by Corollary 2.

Case 5. Assume one component is a tube and the other a basic complex component. Obviously, the tube must be a simple tube. By Lemma 15, the lower and upper parts of the complex component are both simple tubes. Let w be the width of the wall and h its height. The number of S-vertices is $12 + 24 - 2w + 2h = 52$, so we have $h = w + 8$. On the other hand, the number of D-vertices modulo 6 is $2w + (w - 2)h$. By Lemma 16, w is either 2 or 4. For, $w = 4$ the number of D-vertices modulo 6 is 2, a contradiction. Thus, the only possibility is $w = 2$ and $h = 10$. Note that if the tube does not connect (through one or two 0-vertices) to an end of the wall then, a substring $(00H)^9$ is created which does not occur in q. Hence, the tube has to connect to both ends of the wall. Figure 5.25 (a) and (b) depict a schematic view at the connection of the wall and a tube (numbered positions) through one and two 0-vertices, respectively. Notice that if the wall connects to tube through two 0-vertices the first two connections have to be horizontal. If the third connection is vertical then we get the configuration in Figure 5.25(a) in one layer above or below. Clearly, the only way that a tube can be connected to both ends of the wall is when it is in position 2 in Figure 5.25(a). Notice that in this case tube is connected to both ends of wall through one 0-vertex creating an H0H-connection on each end. Furthermore, there will be at least one parallel H0H-connection on each end and in total at least 4 additional H0H-connection, a contradiction.

Case 6. Assume one component is a 2-layer component W, and the other a basic complex component C. We show that the maximum number of (S $\asymp$ h)-connections in this configuration is less than 36. First we count the internal (S $\asymp$ h)-connections in C. All h-vertices in F appear inside the wall and the lower and the upper part of C. The S-vertices of the wall except for the S-vertices on its first and last layers cannot connect to any h-vertex. Therefore, there are at most 4 (S $\asymp$ h)-connections with the S-vertices of the wall. There are at most $12 - w$ S-vertices in the upper (lower) part of C, where w is the wall width. Since $w \geq 2$, there are at most 20 internal (S $\asymp$ h)-connections with the S-vertices of these

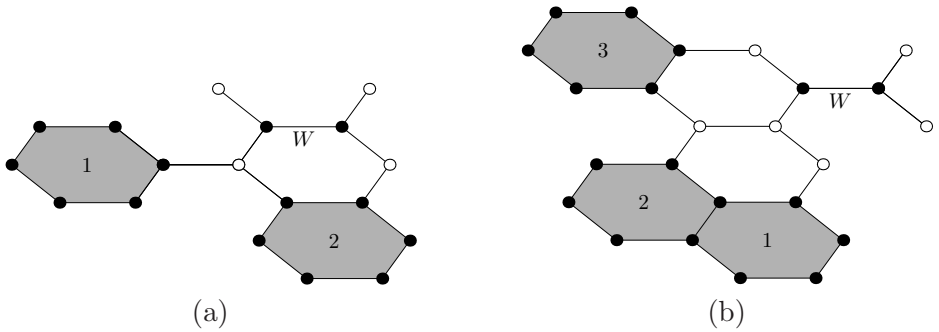


Figure 5.25: A schematic view at the connection of the wall and a tube through one 0-vertex (a) and two 0-vertices (b).

parts. Therefore, there has to be at least 12 external ($\mathsf{S} \asymp \mathsf{h}$)-connections between C and W. It is easy to verify that at most two h-vertices of each side of C can 00-connect to an S-vertex of W. Hence, W has to 00-connect to C from each side. However, one can easily show that for this to happen W must have at least 28 S-vertices in each layer and at least 56 in total, a contradiction.

□

5.4.10 3 components

Lemma 21. *Let F be a saturated fold of q. Then F cannot contain 3 components where none of them is a complex component.*

Proof. Since the second smallest tube has 36 S-vertices, all tubes must be simple. Note that F does not contain a complex component and by Lemma 11, F can contain at most one 2-layer component, hence, to obtain 52 S-vertices, F must have two tubes T_1 and T_2, and a 2-layer component W with two hexagons connected by a path of length 5 in each layer.

By Claim 8, there is no H0H-connection with both ends in W. Therefore, at least one of the tubes, say T_1, must H0H-connect to W. Furthermore, notice that S-vertices of T_1 and T_2 can only provide 24 ($\mathsf{S} \asymp \mathsf{h}$)-connections, so we need to create 12 external ($\mathsf{S} \asymp \mathsf{h}$)-connections between the S-vertices of W and h-vertices of T_1 and T_2. By Claim 7, these connections are either horizontal or vertical. If W and one of the tubes are vertically ($\mathsf{S} \asymp \mathsf{h}$)-connected then we have configuration in Figure 5.26(a). Notice that although in this configuration two ($\mathsf{S} \asymp \mathsf{h}$)-connections are created between the tube and W, we

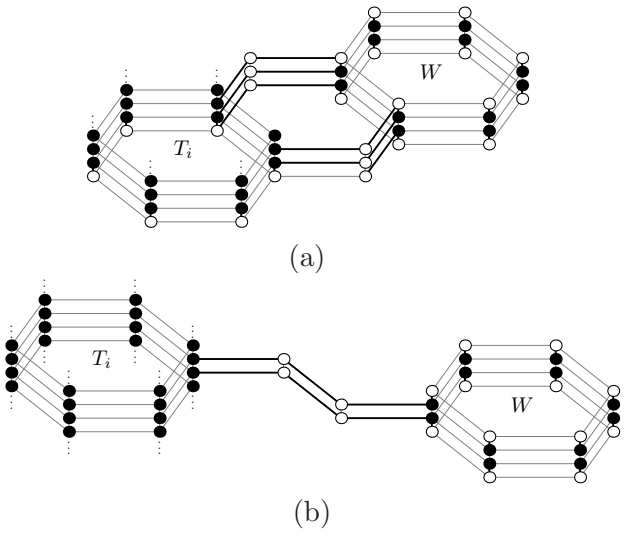


Figure 5.26: Two possible configurations when a tube T_i and a 2-layer component W are $(\mathsf{S} \asymp \mathsf{h})$-connected: with (a) a vertical $(\mathsf{S} \asymp \mathsf{h})$-connection, (b) a horizontal $(\mathsf{S} \asymp \mathsf{h})$-connection.

lose two $(\mathsf{S} \asymp \mathsf{h})$-connections across the tube. Therefore, there are 12 horizontal $(\mathsf{S} \asymp \mathsf{h})$-connections between the tubes and W. The only way to create these connections is depicted in Figure 5.26(b).

Furthermore, since T_1 and W are $\mathsf{H0H}$-connected, by Lemma 12, none of the h-vertices of T_1 is on the same plane as the vh-vertices of W, and hence, they cannot make any horizontal $(\mathsf{S} \asymp \mathsf{h})$-connections. Therefore, all of the 12 $(\mathsf{S} \asymp \mathsf{h})$-connections must be made between W and T_2. This requires that W connects to T_2 from every side which is not possible since the path connecting two hexagons of W has length only 5. □

Lemma 22. *Let F be a saturated fold of q. Then F cannot contain 3 components where one of them is a complex component.*

Proof. Assume that F contains a complex component B. By Lemma 13 and Lemma 17 B is a basic complex component. By Lemma 15, B does not have a 2-layer part. Therefore, the number of S-vertices and the number of D-vertices modulo 6 of B are $24 - 2w + 2h$ and $2w + (w-2)h \bmod 6$, respectively where h is the height and w is the width of the wall of B. By Lemma 16, two values are possible for w: $w = 2$ or $w = 4$. We will consider each case separately.

Case 1. $(w = 4)$ Since F contains at most one 2-layer component, one of the three components in F must be a tube T. Furthermore, B has at least 20 S-vertices, therefore,

the third component can have at most 20 S-vertices. Hence, it can be either another tube T_2, a connector C or a 2-layer component W that consists of two hexagons connected by one edge in each layer. The values for h are 6, 4 and 2 when the third component is T_2, C or W, respectively. For $h = 6, 4$ the number of D-vertices modulo 6 is 2, a contradiction. Therefore, the only possible configuration is the one in which the third component of F is W and $h = 2$.

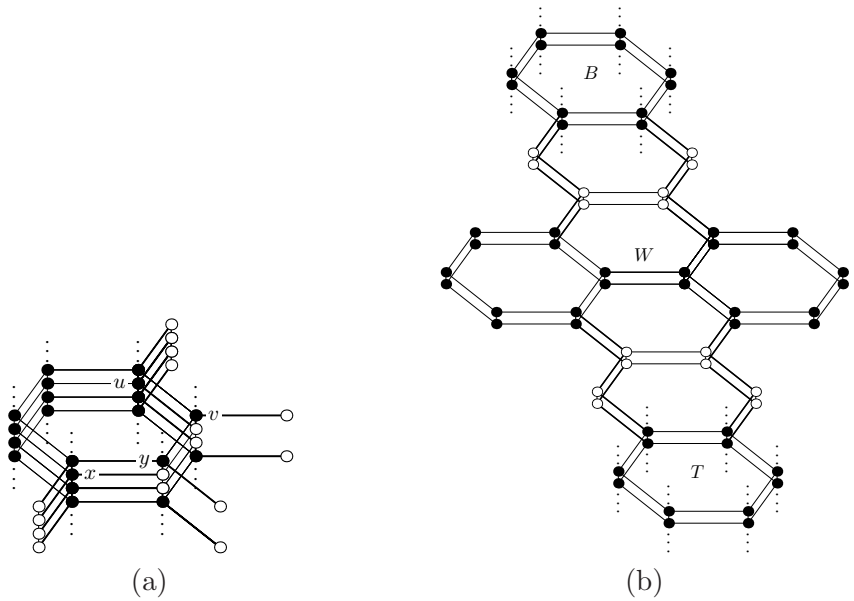


Figure 5.27: (a) A part of a basic complex component with $h = 2$ and $w = 4$. (b) A configuration with a tube T, a 2-layer component W and a basic complex component B.

The basic complex component B is depicted in Figure 5.27(a). It has 20 S-vertices, out of which 8 are part of H0H-connections. Notice that only one of the two S-vertices involved in an H0H-connection (such as x and y) can 00-connect to an h-vertex, otherwise F will contain the substring HH00H0H00HH which does not occure in q. Therefore, the maximum number of possible (S $\asymp$ h)-connections with S-vertices vertices of B and T is $16 + 12 = 28$. Hence, we need to create 8 external (S $\asymp$ h)-connections with the S-vertices of W and h-vertices of B or T. Figure 5.27(b) depicts the only possible configuration to make 8 of such connections. Notice that in this configuration the components are far away to make any H0H-connections with each other so the total number of H0H-connections possible is 4, a contradiction.

Case 2. ($w = 2$) The number of D-vertices modulo 6 of B is 4 independent of the value of

h. Therefore, the only possibility for the other two components in F is that they are both simple tubes, say T and T'. To have right number of S-vertices in F the height h must be 4.

Note that an H-vertex from one side of the wall cannot connect to an H-vertex from the other side of the wall through one or two 0-vertices. Therefore, if the wall is not connected to any vertices of T or T' through one or two 0-vertices, then the two H0H-connections on the same side of wall has to connect through a subsequence containing only S-vertices. This creates a substring which does not occur in q, a contradiction. Therefore, at least one vertex on each side of the wall must connect to a tube.

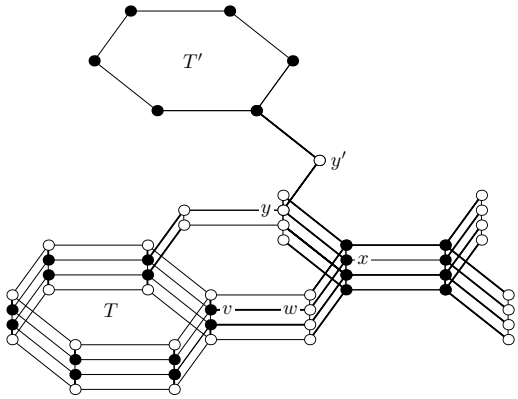
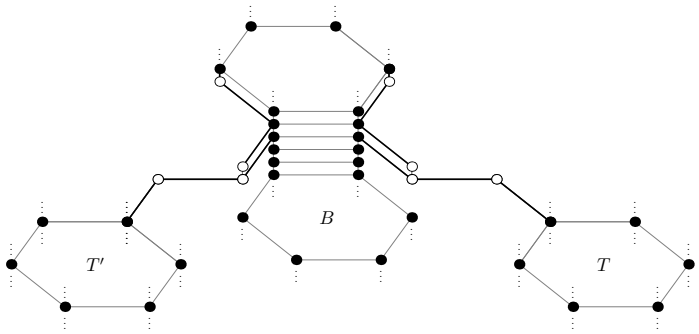


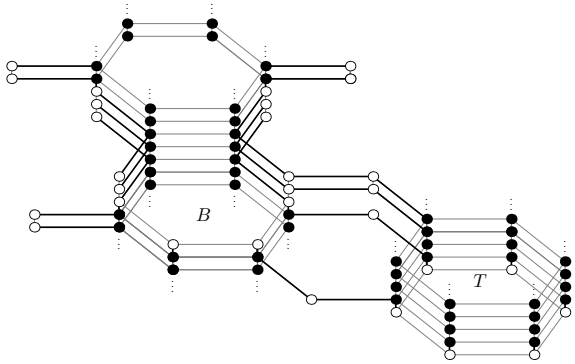
Figure 5.28: H0H-connections between the tube T and a wall of complex component with $w = 2$ and $h = 4$.

First, we show that the wall cannot 0-connect to a tube. To the contrary assume that a vertex v of tube T is connected to a vertex x of the wall through a 0-vertex w. Vertex x cannot be located on the first or the fourth level of the wall otherwise, F would contain the substring H0H0H, a contradiction. Assume that v is in the hexagon that touches that wall. In this case we get another H0H-connection between other side of the wall and T in the same plane. This situation repeats in the plane above or below. Hence, there are at least 4 new H0H-connections, a contradiction. Now, assume that v is not on the hexagon that touches the wall. The vertex v is a vh-vertex otherwise, F would contain a substring H0HH. Without loss of generality assume v^1 is a 0-vertex. One of the vertices v or x must 00-connect to an h-vertex. It is easy to verify that it cannot be v. Therefore, assume that x connects to an h-vertex of T' through 0-vertices y and y'. The only position of T' is shown in Figure 5.28. However, in this configuration the right side of the wall cannot connect to

neither of the tubes, a contradiction. Therefore, each side of the wall is 00-connected to a vertex of a tube.



(a)

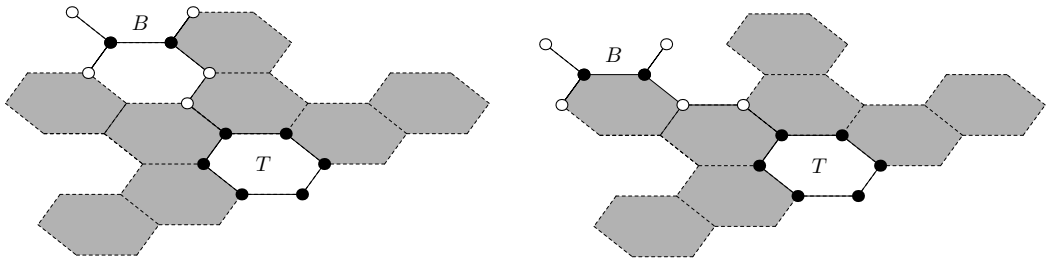


(b)

Figure 5.29: (a) One possible attachment of two tubes to the wall of complex component. (b) H0H-connections of tube T and basic complex component B.

Notice that it is not possible to 00-connect both sides of the wall to the same tube and hence, one side of the wall is 00-connected to T while the other side is 00-connected to T', e.g., Figure 5.29(a).

There are two ways to 00-connect a tube to the wall, cf. Figure 5.30. Note that we need to have two more H0H-connections in F. First, we show that no H0H-connections can be made between B and one of the tubes, say T. Since T cannot H0H-connect to the wall, it would have to connect to the lower or the upper part of B. This is not possible given the relative position of wall of B and T depicted in Figure 5.30(a). If the relative position of the wall of B and T is as depicted in Figure 5.30(b), there is only one possible configuration which is depicted in Figure 5.29(b). However, this configuration contains the substring H0H0H, a contradiction.



(a) (b)

Figure 5.30: Possible configurations of connecting tube T to the wall of the complex component through two 0-vertices. Gray hexagons represent the locations of T' that can H0H-connect to T and is not too far from the wall.

Therefore, the H0H-connections must be made between T and T'. The gray hexagons in Figure 5.30 depicts the possible positions for T'. Clearly, T' cannot 00-connect to the other side of the wall in any of these positions, a contradiction. □

5.4.11 4 components

So far we have proved that any saturated fold F of q must have exactly four components. In this section we prove that the fold F is similar to the designed fold, i.e., that q is structurally stable. First, we show that the components in F are the same as the components in the designed fold.

Theorem 3. *Let F be a saturated fold of q, then F has three simple tubes and a connector. This is true even if the HP model is considered.*

Proof. Since the smallest component other the tube with one hexagon contains at least 16 S-vertices and F contains exactly four components, F must have three simple tubes and one component other than a tube. The three tubes together have 36 S-vertices, therefore, the forth component in F must have 16 S-vertices. The only component with 16 S-vertices is the connector. Therefore, the components in F are the same as the components in the designed fold. □

Next, we prove that in the HPC model the components in F must connect the same way as in the designed fold.

In Lemma 19, two tubes in F can connect with at most two H0H-connections. We will show the same for a tube and connector.

Claim 9. *Let F be a saturated fold of q. A tube and a connector in F can create at most two* H0H*-connections.*

Proof. Assume that the connector C and a tube T are H0H-connected. By Observation 10, this connection is either horizontal or vertical. If the connection is horizontal, by Lemma 12, C and T share only one plane, cf. Figure 5.17(b). Obviously, all other S-vertices of C and T are too far from each other to create more H0H-connections than the two depicted in the figure.

Second, assume there is a vertical H0H-connection between C and T. Then C and T do not share any plane and H0H-connections are created if an edge of C is directly above/below an edge of T. If projections of C and T overlap on more than one edge, then there is a D-vertex of C directly above/below a vertex of T, which would create a substring H0HH in F, a contradiction. Hence, projections of C and T overlap on only one edge, and hence, create exactly two H0H-connections. □

Claim 10. *Let F be a saturated fold of q. Assume that a vertex x of connector C and a a vertex y of a tube T are horizontally ($\mathsf{S} \asymp \mathsf{h}$)-connected in F. Then there are at most two external ($\mathsf{S} \asymp \mathsf{h}$)-connections between them and T is missing at least two internal ($\mathsf{S} \asymp \mathsf{h}$)-connections.*

Proof. By Claim 7, we have the configuration depicted in Figure 5.15(d). Vertex x^{-1} is an S-vertex of C and it cannot be part of a parallel ($\mathsf{S} \asymp \mathsf{h}$)-connection, because y^{-1} is an S-vertex as well. Also note that S-vertex y^{-1} of T cannot be part of internal ($\mathsf{S} \asymp \mathsf{h}$)-connection. Since, horizontal neighbors of y^{-1} and x are H-vertices we have another H0H-connection between these two neighbors and we lose another internal ($\mathsf{S} \asymp \mathsf{h}$)-connection. Similarly, there is at most one ($\mathsf{S} \asymp \mathsf{h}$)-connection between C and T parallel to this H0H-connection. Considering the layout of C and T, it is clear that they cannot ($\mathsf{S} \asymp \mathsf{h}$)-connect at any other point. Hence, the claim follows. □

Observation 12. *Let F be a saturated fold of q. Assume that two tubes T_1 and T_2 are ($\mathsf{S} \asymp \mathsf{h}$)-connected. Then the number of missing internal ($\mathsf{S} \asymp \mathsf{h}$)-connections in T_1 and T_2 minus the number of external ($\mathsf{S} \asymp \mathsf{h}$)-connections between them is at least zero.*

Claim 11. *Let F be a saturated fold of q. Assume that two tubes T_1 and T_2 are* H0H*-connected. Then the number of missing internal ($\mathsf{S} \asymp \mathsf{h}$)-connections in T_1 and T_2 minus the number of external ($\mathsf{S} \asymp \mathsf{h}$)-connections between them is at least two.*

Proof. If T_1 and T_2 are vertically H0H-connected then at most one endpoint of each of two H0H-connections is 00-connected to an h-vertex, since there is no HH00H0H00HH in q. Therefore, we lose at least two internal ($\mathsf{S} \asymp \mathsf{h}$)-connections and gain no external ($\mathsf{S} \asymp \mathsf{h}$)-connections between T_1 and T_2.

If T_1 and T_2 are horizontally H0H-connected we have the configuration depicted in Figure 5.17(b). Vertices x, x', z, z' are S-vertices of the tubes which cannot be part of internal ($\mathsf{S} \asymp \mathsf{h}$)-connections, hence we lose at least four ($\mathsf{S} \asymp \mathsf{h}$)-connections. Furthermore, all possible external ($\mathsf{S} \asymp \mathsf{h}$)-connections between T_1 and T_2 are (x, u, v, w), $(z, z^{-1}, y^{-1}, x^{-1})$, (x', x'^1, y'^1, z'^1) and $(z', z'^{-1}, y'^{-1}, x'^{-1})$. However, first two and last two cannot be present at the same time, otherwise we have HH00H0H00HH in q. Hence, there are at most two such connections. □

Lemma 23. *Let F be a saturated fold of q. The tubes in F have more than 3 layers.*

Proof. Assume that one of the tubes, say T_1, has two or three layers. We prove this lemma by counting the number of possible ($\mathsf{S} \asymp \mathsf{h}$)-connections in F. If T_1 has 2 layers, then it does not contain any internal ($\mathsf{S} \asymp \mathsf{h}$)-connections, since it has no h-vertices. If it has 3 layers then it contains 6 h-vertices, but since they are connected to each other with a peptide bond and there are only two occurrence of substring 0H00HH00H0 in q which occur in the connector, at most one in each pair can be involved in an ($\mathsf{S} \asymp \mathsf{h}$)-connection. Hence, T_1 has at most 3 internal ($\mathsf{S} \asymp \mathsf{h}$)-connections. There should be 36 ($\mathsf{S} \asymp \mathsf{h}$)-connections in F, and the remaining two tubes have at most 24 internal ($\mathsf{S} \asymp \mathsf{h}$)-connections. Hence, F must contain at least 9 external ($\mathsf{S} \asymp \mathsf{h}$)-connections. By Claim 10, any external vertical ($\mathsf{S} \asymp \mathsf{h}$)-connection eliminates at least one internal ($\mathsf{S} \asymp \mathsf{h}$)-connection. Hence, there has to be at least 9 external horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections.

Consider an external horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection (x, u, v, y) connecting components W_1 and W_2, cf. Figure 5.15(c). By Lemma 19 and Claim 9, any pair of components in F can create at most two H0H-connections, i.e, at least three pairs of components are H0H-connected. Since these pairs cannot be horizontally ($\mathsf{S} \asymp \mathsf{h}$)-connected, there are at most three pairs of horizontally ($\mathsf{S} \asymp \mathsf{h}$)-connected components. Hence, by Claim 2, there are at most 6 horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections, a contradictions. □

We proceed by proving the following lemma.

Lemma 24. *Let F be a saturated fold of q. Any component in F must be* H0H*-connected to at least one other component.*

Proof. By Lemma 18 and Claim 9, there are at most two H0H-connections between any two components of F. Since F contains 6 H0H-connections it is enough to show that there is no cycle of length 3 of H0H-connected components. Let components W_1,W_2,W_3 form such a cycle. By Lemmas 12 and 19 and Claim 9, two H0H-connected components are either in the configuration depicted in Figure 5.17(b) or Figure 5.17(c), i.e., they share exactly one plane or they share no planes and there is one plane in between them. Assume that W_1 is the topmost component in planes $H_i, H_{i+1}, \ldots, H_j$. If both W_2 and W_3 share one plane with W_1 (or none of them share any plane with W_1) then they share at least two layers, i.e., they cannot be H0H-connected. Hence, assume that W_2 shares plane H_i with W_1, i.e., it is located in planes $H_i, H_{i-1}, \ldots$ and W_3 does not share any plane, i.e., it is located in planes $H_{i-2}, H_{i-3}, \ldots$. Then W_2 and W_3 can share zero or one plane only if W_2 has either one or three layers. Obviously, the first case is not possible. In the second case, W_2 must be a tube, but by Lemma 23, it cannot have 3 layers, a contradiction. □

We proceed by proving the following important lemma:

Lemma 25. *Let F be a saturated fold of q. Two tubes in F cannot be* H0H*-connected. Consequently, two tubes cannot be vertically* ($\mathsf{S} \asymp \mathsf{h}$)*-connected.*

Proof. To the contrary assume that two tubes are H0H-connected. By Claim 10, we need at least two external ($\mathsf{S} \asymp \mathsf{h}$)-connections, and by Claim 10 and Observation 12, there are at most two horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections between the connector C and a tube T_1.

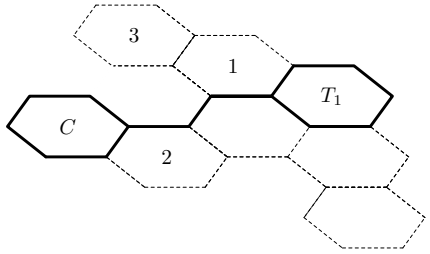


Figure 5.31: The schematic view at horizontally ($\mathsf{S} \asymp \mathsf{h}$)-connected connector C and tube T_1. The numbers show all possible locations of tube T_2 which is H0H-connected to both T_1 and C.

Figure 5.31 shows a schematic view at the horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections between T_1 and C. Notice that T_1 and C cannot be H0H-connected. Therefore, by Lemma 24, T_1 must

H0H-connect to another tube T_2. We will show that T_2 cannot be H0H-connected to C. Assume the contrary. The tube T_2 must be located in one of the three numbered positions in Figure 5.31.

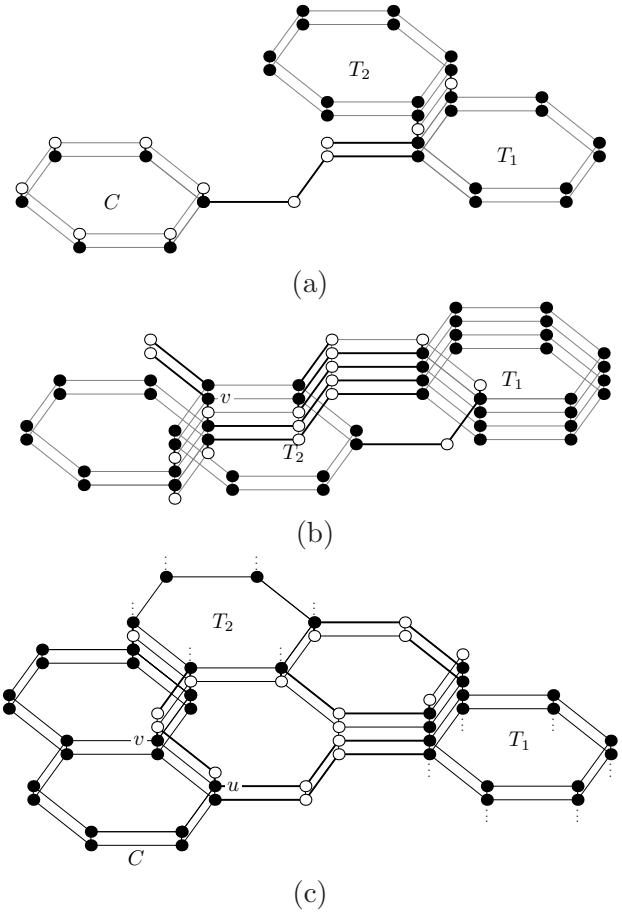


Figure 5.32: Three possible configurations when connector C is horizontally ($\mathsf{S} \asymp \mathsf{h}$)-connected to T_1, and T_2 is H0H-connected to both C and T_1.

Figure 5.32 depicts configurations for all three positions of T_2. Clearly, in the first configuration T_2 cannot make any H0H-connections with C (cf. Figure 5.32(a)). Consider vertex v in Figure 5.32(b) depicting the second configuration. It is H0H-connected to v^{-2} which is part of an ($\mathsf{S} \asymp \mathsf{h}$)-connection. Since there is no substring HH00H0H00HH in q, v cannot be part of any ($\mathsf{S} \asymp \mathsf{h}$)-connection. Therefore, we lose one more internal ($\mathsf{S} \asymp \mathsf{h}$)-connection in T_2 which needs to be replaced by an external ($\mathsf{S} \asymp \mathsf{h}$)-connection between C and a tube. By Claim 10, any external vertical ($\mathsf{S} \asymp \mathsf{h}$)-connection eliminates at least

one internal ($\mathsf{S} \asymp \mathsf{h}$)-connection, therefore, the replaced connection must be a horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection. Clearly, T_2 cannot make any horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections with C and furthermore, T_1 cannot make any new horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections with C. Hence, T_3 must make at least one horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection which in this case T_3 cannot be $\mathsf{H0H}$-connected to C. Therefore, T_3 must $\mathsf{H0H}$-connect to T_1 or T_2. In this case we lose at least two additional internal ($\mathsf{S} \asymp \mathsf{h}$)-connections which cannot be replaced by any external horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections. Finally, we show that the third configuration is contradictory. Consider the v-vertex v in Figure 5.32(c). If it is $\mathsf{00}$-connected to w or x it follows that v is a part of the substring $(\mathsf{00HH})^k$, a contradiction by Lemma 11. Therefore, v is $\mathsf{00}$-connected to u. However in this case, F contains the substring $\mathsf{HH00H00HH}$ which does not occur in q, a contradiction.

It follows that T_2 and C are not $\mathsf{H0H}$-connected. Therefore, by Lemma 24, T_3 must $\mathsf{H0H}$-connect to C and to have 6 $\mathsf{H0H}$-connections in F, T_3 must also $\mathsf{H0H}$-connect to T_1 or T_2. However, in this case we lose at least two additional internal ($\mathsf{S} \asymp \mathsf{h}$)-connections which by Claim 10, must be replaced by horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections between C and a tube. Clearly, T_3 cannot make such connections with C. Furthermore, T_1 cannot make new horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections with C. Thus, T_2 must make two horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections with C. Let H_i and H_{i+1} be the layers of C. Without loss of generality assume that T_2 is above T_1. Since C and T_1 make horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connections the top most layer H_j of T_1 is above H_{i+1}. Let H_l be be the lowest layer of T_2. Since T_1 and T_2 are $\mathsf{H0H}$-connected, $l \geq j > i+1$. Therefore, C and T_2 do not share any layer and hence, cannot be ($\mathsf{S} \asymp \mathsf{h}$)-connected, a contradiction. □

Corollary 5. *Let F be a saturated fold of q. All tubes in F must be* $\mathsf{H0H}$*-connected to the connector.*

Proof. We consider three cases. If the connection is between two h-vertices then clearly all edges of the connection must be horizontal. Second the case when the connection is between h- and S-vertices follows by Lemma 25. Finally, if the connection is between two S-vertices, we lose two internal ($\mathsf{S} \asymp \mathsf{h}$)-connections which can be only replaced by horizontal ($\mathsf{S} \asymp \mathsf{h}$)-connection between connector and a tube. By Corollary 5, this is not possible. □

So far we have shown that all tubes must $\mathsf{H0H}$-connect to C. We prove the final theorem.

Theorem 4. *The protein string q is structurally stable.*

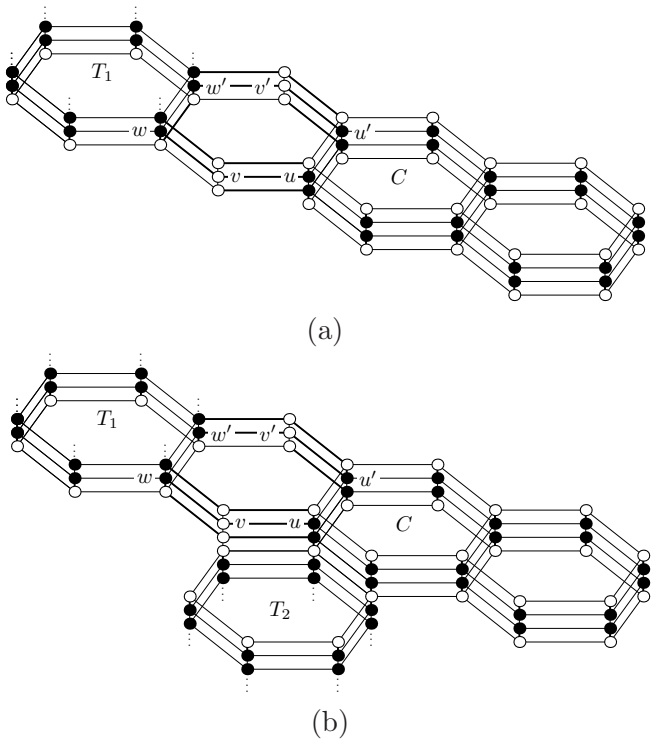


Figure 5.33: Two possible configurations that contain the substring $t = \mathsf{10100102002}$, given that one of the H0H-connections in t is horizontal.

Proof. Let F be a saturated fold of q. By Theorem 3 and Corollary 5, F contains three simple tubes which are H0H-connected to a connector C. Note that there are no H0H-connections between tubes. Note that F must contain one occurrence of the substring $t = \mathsf{10100102002}$. The substring t contains two H0H-connections that are 00-connected. We show that these H0H-connections are vertical and they belong to two tubes T_1 and T_2 where T_1 is connected to the top and T_2 is connected to the bottom of C. To the contrary, assume that one of the H0H-connections (u, v, w) in t is horizontal, where u and w are H-vertices in C and T_1, respectively, and v is a 0-vertex. Note that C and T_1 make another H0H-connection (u', v', w') where u' and v' are horizontal neighbors of u and v respectively. Vertex u or w (respectively, u' or w') must 00-connect to an h-vertex. It is easy to see that w (w') cannot 00-connect to an h-vertex and the only h-vertex that u (u') can 00-connect to is w^1 (w'^1). Therefore, w must 00-connect to an H0H-connection.

Two configurations are possible in this case. In the first configuration w is 00-connected to w', cf. Figure 5.33(a), and hence, exactly one of the pairs of vertices (u, w^1) or (u', w'^1)

contains 2-vertices. Since T_1 makes H0H-connections only with C and every 2-vertex is either a part of H0H-connection or is 00-connected to an H0H-connection, w^1 (respectively, w'^1) cannot be paired with a 2-vertex, a contradiction. In the second configuration w is 00-connected to u^{-1} and C is vertically H0H-connected to another tube T_2 at u^{-1} and its horizontal neighbor (cf. Figure 5.33(b)). Note that T_3 must connect to the hexagon of C that does not contain u and u^{-1} otherwise, F would contain the substring $(00H)^6$, a contradiction. Therefore, T_1 is too far from T_2 and T_3 to 00-connect to either of them. Hence, w^1 is p-connected to w'^1 by a path p which lies completely in T_1 and its 0-vertices (0-vertices surrounding T_1). Consequently, p does not contain any H0H as a substring. Since H0H-connection (w, v, u) is 00-connected to H0H-connection (u^{-1}, u^{-2}, u^{-3}), based on the properties of q, it follows that exactly one of the pairs $(u,\ w^1)$ or $(u',\ w'^1)$ contains 2-vertices, depending on the direction of the substring t. Clearly, w^1 (respectively, w'^1) cannot be paired with any other 2-vertex, a contradiction. Therefore, both H0H-connections in t are vertical.

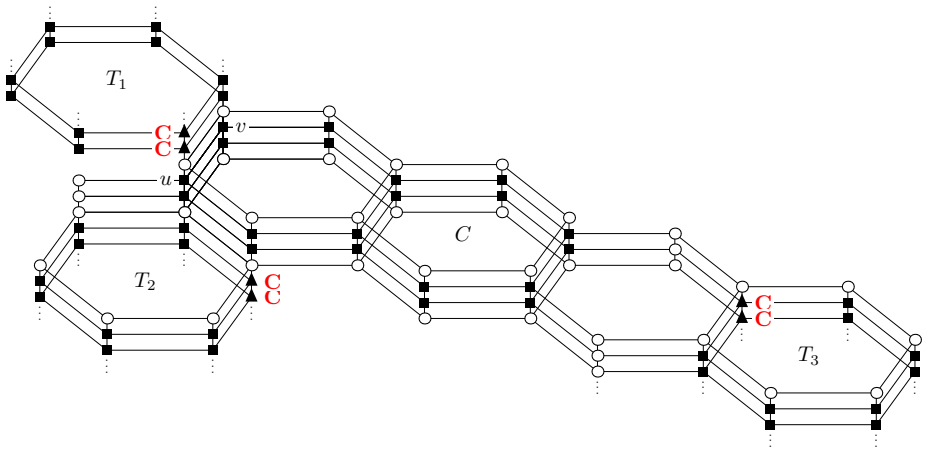


Figure 5.34: The only possible configuration that contains the substring $t = 10100102002$, given that the H0H-connections in t are vertical.

Let (u, u^1, u^2) be one of the H0H-connections in t where u and u^2 are H-vertices in C and T_1, respectively, and u^1 is a 0-vertex. Without loss of generality assume that T_1 is connected to the top of C. Note that T_1 and C make another vertical H0H-connection (v, v^1, v^2), where v is a horizontal neighbor of u. Clearly, u cannot 00-connect to an h-vertex, therefore, it must 00-connect to another vertical H0H-connection. The only possibility is that u is 00-connected to u^{-1}. Therefore, C vertically H0H-connect to another tube T_2 at the vertex u^{-1}

and one of its horizontal neighbors. If this second connection is (v^{-1}, v^{-2}, v^{-3}) then F would contain another occurrence of the substring t through vertices $v^2, v^1, v, *, *, v^{-1}, v^{-2}, v^{-3}$, a contradiction. It follows that T_1 and T_2 are H0H-connected to C as in the original fold. It is easy to see that the last tube T_3 must horizontally H0H-connect to the other side of C as in the original fold (cf. Figure 5.34).

Finally, notice that T_1, T_2 and T_3 are far away from each other to make any 00-connections. Therefore, the pair of H0H-connections in each tube are p-connected by a path p that lies completely in that tube and its 0-vertices. This implies that the length of the tubes must be the same as the length of the tubes in the original fold and hence, q is structurally stable. □

Chapter 6

Conclusions and future works

6.1 Conclusions

In this dissertation we study the design of robust classes of stable proteins under 2D and 3D HP model. The HP model was introduced by Dill [41] over two decades ago and the protein folding problems (forward and inverse) have been studied under this model since. We extended this model by using a third type of amino acid, the cysteine, in the designed proteins and incorporating the disulfide bridges between two cysteines in the energy model. One of the interesting problems in protein design initiated by Gupta el al [60] is the design of classes of stable proteins that can *approximate* target shapes. In this thesis we solve this problem in 2D square lattice and introduce a robust class of 3D protein structures, called tubular structures, and give evidence that they are stable under the HPC model. In particular we prove that an infinite class of basic tubular structures consisting of a connector and three tubes of arbitrary lengths are stable under the HPC model.

The linear constructible structures were proposed in [60] as a good candidate for proving their stability. We introduce two robust subclasses of linear constructible structures, the snake and wave structures, that are capable of approximating any given 2D shape. We refine these structures for the HPC model and proved that the proteins of snake structures are stable under an artificial variant of the HPC model called the strong HPC model and that the proteins of wave structures are stable under the biologically motivated HPC model. The stability proof of the wave structures partially confirms the conjecture stated in [60].

We extend the results in 2D and solve the shape-approximating inverse protein folding problem under the HP model in 3D. We design tubular proteins using two basic building

blocks: a tube and a connector. These blocks can be interconnected to roughly approximate any given shape. We showed that a simple subclass of the structures built in this way is structurally stable in the HPC model. Showing that all these structures are structurally stable is a very challenging problem. The first task in solving this problem is to choose which of the hydrophobic monomers are cysteines. The second is to prove that all folds are similar to the designed one. This gets more difficult with the higher number of building blocks (tubes and connectors) used, as each additional building block adds two particular substrings to the protein sequence, which increase the variety and the number of possible components in the fold.

While the techniques presented here will not allow for the direct construction of proteins, they represent a starting point for this process. In particular, we believe that these techniques can be used to form the basis of an actual protein — we specify, at each point of the chain whether a cysteine, other hydrophobic or polar monomer is required and a designer can use this information to choose amino acids from set of all 20 amino acids. The choice of actual amino acid would depend on other desired molecular interactions and finer details about the protein structure.

6.2 Future works

The complexity of the inverse protein folding problem is a long standing open problem in proteomics. Other complexity questions related to forward protein folding are (a) complexity of determining whether the sequence is stable; (b) whether it has a saturated fold (each hydrophobic amino acid has the maximum possible number of contacts); (c) whether a contact map is realizable in a lattice used; etc. Answers to these questions could help us building computational tools for simplifying and extending the designs we introduce in this dissertation. As a continuation to our work in 2D and 3D lattices, we are interested in investigating designing tiles for various types of lattices (mainly 3D) which can be used to build stable proteins approximating any given shape. Such successful designs could be later extended to actual sequences of amino acids (replacing H and P letters with amino acids) based on the database of bond and torsion angles of C-alpha atoms in backbones of existing proteins.

Other motivated problems are proving the stability proof of more general classes of constructible structures and tubular structures.

Protein interactions often exhibit a "hand-in-glove" fit since this facilitates specific chemical reactions and helps to build protein complexes. By these means, chemically assembled proteins with appropriate surface and binding properties can attach to specific targets and modify their functions. In protein-drug interaction designs, we are interested in exactly such surface properties of synthetic proteins. An interesting research direction is to apply our techniques for IPF to this new problem and aid in the design of proteins that will bind to a specific target. An intermediate goal is to understand the dynamics of the problem in the lattice context and to design stable binding proteins for a class of simple targets in the 2D and 3D HP models.

Appendix A

2DHPSolver pseudo code

Input: designRules, initialFiled, depth

vector FIELD Fields;

Fields.insert(initialField);

activeFiled=0;

while *Fileds.size()* > 0 **do**

    display(Fileds, activeField);

    userExtension=getUserExtension();

    applyExtension(Fields, activeField, userExtension, designRules, depth);

end

Algorithm 1: 2DHPSolver

```
Input: userExtension, Fields, activeField, designRules, depth
if userExtension.type = change activeField then
    activeField = get a new index;
    return;
end
FIELD current=Fields.remove(activeField);
foreach possible userExtension.type E at userExtension.point P do
    newField = apply E at P to current;
    if newField does not violate the designRules then
        (result, resultField)=selfExtend(newField, designRules, depth,
        maxExtension);
        if result ≠ DELETE then
            Fields.insert(resultField);
        end
    end
end
```

Algorithm 2: applyUserExtension

```
Input: field, designRules, depth, maxExtension
Output: result, extendedField
changed=FALSE;
for i=1 to maxExtension do
    foreach possible extension point P of field do
        (result, tmpField)=selfExtendAtOnePoint(field, P, designRules, depth,
        maxExtension);
        if result=EXTEND then
            changed=TRUE;
            field=tmpField;
            break;
        end
        if result=DELETE then
            return;
        end
    end
end
extendedField=field;
if changed=TRUE then
    result=EXTEND;
else
    result=NOCHANGE;
end
return
```

Algorithm 3: selfExtend

```
Input: field, P, depth, designRules
Output: result, extendedField
validFields=0;
foreach possible extension E at point P do
    tmpField = apply E at P to field;
    if depth = 1 then
        if tmpField does not violate designRules then
            if validFields > 0 then
                result=NOCHANGE;
                return;
            end
            validFields ++;
            extendedField=tmpField;
        end
    else
        (tmpResult, tmpExtendedField)=selfExtend(tmpField, designRules, depth-1,
        maxExtension);
        if tmpResult ≠ DELETE then
            if validFields > 0 then
                result=NOCHANGE;
                return;
            end
            validFields ++;
            extendedField=tmpExtendedField;
        end
    end
end
if validFields = 0 then
    result=DELETE;
else
    result=EXTEND;
end
```

Algorithm 4: selfExtendAtOnePoint

Appendix B

Snake's forbidden subsequences

0: Polar, 1: Hydrophobic, R: Cysteine, B: Hydrophobic non-cysteine

11

1010101

00100100100

000

R0R

B0B

B00R

R00B

10B01

B00B00

00B00B

10R00R01

1010100101010010101

00100101001010010010100100100

00100100 this subsequence occurs at most twice in a snake structure.

001010010100101

101001010010100

10101001010010010010100101010

00101001010010010010100101010

Appendix C

Wave's forbidden subsequences

11
000
R0R
B0B
B00R
10B01
B00B00
1010101
00100100100
10R00R01
00100100 this subsequence occurs at most twice in a wave structure.
1010100101010010101
10100100101001010010010100101001001010
001010010100101
10101001010010010010100101
10010100101001001001010010100101001
00R00R00R0B00B0R0B00B0R00
00R00R00R0B00B0R00R00R0B00B0R0B
B0R0B00B0R0B00B0R00R00R0B00B0R00
B0R0B00B0R00R00R0B00B0R0B00B0R00R00R0B00B0R0B
B0R00R00R0B00B0R0B00B0R00R00R0B00B0R0B00B0R00
B0R00R00R0B00B0R00R00R0B00B0R0B00B0R00R00R0B00B0R00

Bibliography

[1] O. Aichholzer, D. Bremner, E. Demaine, H. Meijer, V. Sacristán, and M. Soss. Long proteins with unique optimal foldings in the H-P model. *Computational Geometry: Theory and Applications*, 25(1-2):139–159, 2003.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garlan Science, 2002.

[3] B. Allen and S. Mayo. Dramatic performance enhancements for the faster optimization algorithm. *J Comput Chem.*, 27(10):1071–1075, 2006.

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

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

[6] C. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–238, 1973.

[7] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *J. Comp. Biol.*, 5(1):27–40, 1998.

[8] P. Berman, B. DasGupta, D. Mubayi, R. Sloan, Gyorgy, Turan, , and Y. Zhang. The inverse protein folding problem on 2d and 3d lattices. *Discrete Applied Mathematics*, 155:719–732, 2007.

[9] P. Berman, B. DasGupta, D. Mubayi, R. Sloan, G. Turán, and Y. Zhang. The inverse protein folding problem on 2D and 3D lattices. *Discr. Appl. Math.*, 155:719–732, 2007.

[10] C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing, Incorporated, 1999.

[11] J. Bryngelson, J. Onuchic, N. Socci, and P. Wolynes. Funnels, pathways and the energy landscape of protein folding: A synthesis. *PROTEINS: Structure, Function and Genetics*, 21:167–195, 1995.

[12] G. L. Butterfoss and B. Kuhlman. Computer-based design of novel protein structures. *Annu. Rev. Biophys. Biomol. Struct.*, 35:49–65, 2006.

[13] E. Buxbaum. *Fundamentals of protein structures and function*. Springer, 2007.

[14] C. Camacho and D. Thirumalai. Kinetics and thermodynamics of folding in model proteins. In *Proc. Natl. Acad. Sci. USA*, volume 90, pages 6369–6372, 1993.

[15] A. Campbell and L. Heyer. *Discovering genetics, proteomics, and bioinformatics*. Pearson Education, 2007.

[16] T. Cardozo, M. Totrov, and R. Abagyan. Homology modeling by the icm method. *Proteins: Structure, Function and Genetics*, 23:403–414, 1995.

[17] J. Cavanagh, W. J. Fairbrother, A. Palimer, M. Rance, and N. Skelton. *Protein NMR Spectroscopy: Principles and Practice*. Elsevier Academic Press, 2007.

[18] H. Chan and K. Dill. "sequence space soup" of proteins and copolymers. *Journal of Chemical Physics*, 95:3775–3787, 1991.

[19] H. Chan and K. Dill. The protein folding problem. *Physics Today*, 46:24–32, 1993.

[20] H. Chan and K. Dill. A simple model of chaperonin-mediated protein folding. *Proteins: Structure, Function, and Genetics*, 24:345–351, 1996.

[21] B. Chazelle, C. Kingsford, and M. Singh. A semidefinite programming approach to side-chain positioning with new round ing strategies. *INFORMS J. Computing*, 16:380–392, 2004.

[22] C. Chothia and A. Lesk. The relation between the divergence of sequence and structure in proteins. *EMBO Journal*, 5:823–836, 1986.

[23] K. Chou, A. Heckel, G. Nmethy, S. Rumsey, L. Carlacci, and H. Scheraga. Energetics of the structure and chain tilting of antiparallel beta-barrels in proteins. *Proteins.*, 8(1):14–22, 1990.

[24] K. Chou, M. Pottle, G. Nmethy, Y. Ueda, and H. Scheraga. Structure of beta-sheets. origin of the right-handed twist and of the increased stability of antiparallel over parallel sheets. *Journal of Molecular Biology*, 162(1):89–112, 1982.

[25] P. Chowdhury, G. Vasmatzis, B. Lee, and I. Pastan. Improved stability and yield of a fv-toxin fusion protein by computer design and protein engineering of the fv. *Journal of Molecular Biology*, 281:917–928, 1998.

[26] N. Clarke and S. Yuan. Metal search: a computer program that helps design tetrahedral metal-binding sites. *Proteins*, 23(2):256–63., 1995.

[27] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. In *Proc. of STOC'98*, pages 597–603, 1998.

[28] J. C. Daan and R. Sindelar. *Pharmaceutical Biotechnology : an Introduction for Pharmacists and Pharmaceutical Scientists*. Informa Health Care, 2002.

[29] B. Dahiyat, D. Gordon, and S. Mayo. Automated design of the surface positions of protein helices. *Protein Sci.*, 6:1333–1337, 1997.

[30] B. Dahiyat and S. Mayo. Protein design automation. *Protein Sci.*, 5:895–903, 1996.

[31] B. Dahiyat and S. Mayo. De novo design: fully automated sequence selection. *Science*, 278:82–87, 1997.

[32] B. Dahiyat and S. Mayo. Probing the role of packing specificity in protein design. *Proc Natl Acad Sci*, 94:10172–10177, 1997.

[33] J. R. D.C. and Richardson. The de novo design of protein structures. *Trends Biochem. Sci.*, 14:304–309, 1989.

[34] W. DeGrado, Z. Wasserman, and J. Lear. Protein design, a minimalist approach. *Science*, 243:622–628, 1989.

[35] J. Desjarlais and N. Clarke. Computer search algorithms in protein modification and design. *Curr. Opin. Struct. Biol.*, 101:471–475, 1998.

[36] J. Desjarlais and T. Handel. De novo design of the hydrophobic cores of proteins. *Protein Sci*, 4:2006–2018, 1995.

[37] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, 1992.

[38] J. Desmet, M. D. Maeyer, and I. Lasters. The dead-end elimination theorem: a new approach to the side-chain packing problem. *The Protein Folding Problem and Tertiary Structure Prediction*, 91:307–337, 1994.

[39] J. Desmet, J. Spriet, and I. Lasters. Fast and accurate side-chain topology and energy refinement (faster) as a new method for protein structure optimization. *Proteins: Structure, Function, and Bioinformatics*, 48:31–43, 2002.

[40] J. M. Deutsch and T. Kurosky. New algorithm for protein design. *Physical Review Letters*, 76:323–326, 1996.

[41] K. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, 1985.

[42] K. Dill, S. Bromberg, K. Yue, K. Fiebig, D. Yee, P. Thomas, and H. Chan. Principles of protein folding: A perspective from simple exact models. *Protein Science*, 4:561–602, 1995.

[43] C. Dobson and A. Fersht. *Protein Folding*. Cambridge University Press, 1996.

[44] J. Drenth. *Principles of Protein X-ray Crystallography*. Springer, 1999.

[45] Y. Duan and P. Kollman. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution . *Science*, 282:740–744, 1998.

[46] R. Dunbrack and M. Karplus. Backbone-dependent rotamer library for proteins: Application to side-chain prediction. *Journal of Molecular Biology*, 230:543–574, 1993.

[47] M. Dwyer, L. Looger, and H. Hellinga. Computational design of a biologically active enzyme. *Science*, 304:1967–1971, 2004.

[48] S. Edinger, C. Cortis, P. Shenkin, and R. Friesner. Solvation free energies of peptides: Comparison of approximate continuum solvation models with accurate solution of the poisson-boltzmann equation. *J. Phys. Chem. B*, 101:1190–1197, 1997.

[49] I. Eidhammer, I. Jonassen, and W. Taylor. *Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis*. John Wiley and Sons, 2004.

[50] D. Eisenberg and A. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319:199–203, 1986.

[51] D. Eisenberg, W. Wilcox, S. Eshita, P. Pryciak, S. Ho, and W. DeGrado. The design, synthesis, and crystallization of an alpha-helical peptide. *Proteins*, 1(1):16–22, 1986.

[52] X. Fu, H. Kono, and J. Saven. Probabilistic approach to the design of symmetric protein quaternary structures. *Protein Eng. Des*, 16(12):971–977, 2003.

[53] I. Georgiev, R. Lilien, and B. Donald. A novel minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. *Lecture Notes in Computer Science.*, 3909:530–545, 2006.

[54] B. Gibney, F. Rabanal, and P. Dutton. Synthesis of novel proteins. *Curr. Opin. Chem. Biol.*, 1(4):537–542, 1997.

[55] A. Godzik. In search of the ideal protein sequence. *Protein Eng.*, 8:409–416, 1995.

[56] R. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.*, 66:1335–1340, 1994.

[57] D. Gordon, G. Hom, and S. Mayo. Exact rotamer optimization for protein design. *J. Comput. Chem.*, 24:232–243, 2003.

[58] D. Gordon, S. Marshall, and S. Mayo. Energy functions for protein design. *Curr. Opin. Struct. Biol.*, 16(12):509–513, 1999.

[59] D. Gordon and S. Mayo. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comput. Chem.*, 19:1505–1514, 1998.

[60] A. Gupta, J. Maňuch, and L. Stacho. Structure-approximating inverse protein folding problem in the 2D HP model. *Journal of Computational Biology*, 12(10):1328–1345, 2005.

[61] T. Hansson, C. Oostenbrink, and W. van Gunsteren. Molecular dynamics simulations. *Current Opinion in Structural Biology.*, 12:190–196, 2002.

[62] P. Harbury, B. Tidor, and P. Kim. Energetics of the structure and chain tilting of antiparallel beta-barrels in proteins. *Proc. Natl. Acad. Sci. USA*, 92:8408–8412, 1995.

[63] P. Harbury, B. Tidor, and P. Kim. Repacking protein cores with backbone freedom: structure prediction for coiled coils. *Proc Natl Acad Sci*, 92:8408–8412, 1995.

[64] C. Hardina, T. V. Pogorelovb, and Z. Luthey-Schultena. Ab initio protein structure prediction. *Current Opinion in Structural Biology.*, 12:176–181, 2002.

[65] R. Harrison, D. Chatterjee, and I. Weber. Analysis of six protein structures predicted y comparative modeling techniques. *Proteins: Structure, Function and Genetics*, 23:463–471, 1995.

[66] W. Hart. On the computational complexity of sequence design problems. In *Proc. of Comp. Molecular Biology*, pages 128–136, 1997.

[67] T. Havel and M. Snow. A new method for building protein conformations from sequence alignments with homologous of known structure. *Journal of molecular biology*, 217:1–7, 1991.

[68] B. Hayes. Prototeins. *American Scientist*, 86:216–221, 1998.

[69] H. Hellinga and F. Richards. Optimal sequence selection in proteins of known structure by simulated evolution. *Proc Natl Acad Sci*, 91:5803–5807, 1994.

[70] H. W. Hellinga and F. M. Richards. Construction of new ligand binding sites in proteins of known structure i. computer-aided modeling of sites with pre-defined geometry. *Journal of Molecular Biology*, 222:763–785, 1991.

[71] S. Ho and W. DeGrado. Design of a 4-helix bundle protein: synthesis of peptides which self-associated into a helical protein. *Journal of the American Chemical Society*, 109:6751–6758, 1987.

[72] J. Holland. Adaptation in natural and artificial systems. *The MIT Press, Boston*, 1993.

[73] R. Jaenicke. Protein stability and molecular adaptation to extreme conditions. *Eur. J. Biochem.*, 202:715–728, 1991.

[74] J. Janin, S. Wodak, M. Levitt, and D. Maigret. The conformation of amino acid side chains in proteins. *Journal of Molecular Biology*, 125:357–386, 1978.

[75] A. Jaramillo, L. Wernisch, S. Hry, and S. J. Wodak. Automatic procedures for protein design. *Combinatorial Chemistry and High Throughput Screening*, 4:643–659, 2001.

[76] T. Jiang and L. Wang. Algorithmic methods for multiple sequence alignment. *Current Topics in Computational Molecular Biology*, 2002.

[77] P. Jolls and H. Jrnvall. *Proteomics in Functional Genomics: Protein Structure Analysis*. Birkhuser, 2000.

[78] D. Jones. De novo protein design using pairwise potentials and a genetic algorithm. *Protein Sci.*, 3:567–574, 1994.

[79] D. Jones. Predicting novel protein folds by using fragfold. *Proteins: Structure, Function and Genetics*, Supplement 5:127–132, 2002.

[80] T. Jones and S. Thriup. Using known substructures in protein model building and crystallography. *EMBO Journal*, 5:819–823, 1986.

[81] H. Kadokura. Oxidative protein folding: Many different ways to introduce disulfide bonds. *Antioxidants and Redox Signaling.*, 8:731–733, 2006.

[82] S. Kamtekar, J. Schiffer, H. Xiong, J. Babik, and M. Hecht. Protein design by binary patterning of polar and nonpolar amino acids. *Science*, 262:1680–1685, 1993.

[83] I. Karle, S. Awasthi, and P. Balaram. A designed beta-hairpin peptide in crystals. *Proc Natl Acad Sci USA*, 93(16):8189–8193, 1996.

[84] J. Kleinberg. Efficient algorithms for protein sequence design and the analysis of certain evolutionary fitness landscapes. In *Proc. Third Annual International Conference on Computational Molecular Biology*, pages 226–237, 1999.

[85] P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.*, 239:249–275, 1994.

[86] P. Koehl and M. Levitt. De novo protein design. i. in search of stability and specificity. *Journal of Molecular Biology*, 293:1161–1181, 1999.

[87] A. Kolinski and J. Skolnick. Reduced models of proteins and their applications. *Polymer*, 45:6511–524, 2004.

[88] C. M. Kraemer-Pecore, A. M. Wollacott, and J. R. Desjarlais. Computational protein design. *Current Opinion in Chemical Biology*, 5:690–695, 2001.

[89] R. Kreitman and I. Pastan. Immunotoxins for targeted cancer-therapy. *Advan. Drug Deliv. Rev.*, 31:53–88, 1998.

[90] B. Kuhlman, G. Dantas, G. Ireton, G. Varani, B. Stoddard, and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302:1364–1368, 2003.

[91] B. Kuhlman, G. Dantas, G. Ireton, G. Varani, B. Stoddard, and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–1368, 2003.

[92] T. Kurosky and J. Deutsch. Design of copolymeric materials. *Physics A: Mathematical and General*, 28:387–393, 1995.

[93] T. Kurosky and J. M. Deutsch. Design of copolymeric materials. *J. Phys. A.*, 27:L387–L393, 1995.

[94] E. Lacroix, T. Kortemme, M. L. de la Paz, and L. Serrano. The design of linear peptides that fold as monomeric beta-sheet structures. *Curr. Opin. Struct. Biol.*, 9:487–493, 1999.

[95] R. Laskowski, M. MacArthur, D. Moss, and J. Thornton. Procheck: a program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography*, 26:283–291, 1993.

[96] K. Lau and K. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22:3986–3997, 1989.

[97] K. Lau and K. Dill. Theory for protein mutability and biogenesis. *Proc. Natl. Acad. Sci. U.S.A.*, 87:6388–6392, 1990.

[98] G. Lazar, J. Desjarlais, and T. Handel. De novo design of the hydrophobic core of ubiquitin. *Protein Sci.*, 6:1167–1178, 1997.

[99] C. Lee. Predicting protein mutant energetics by self-consistent ensemble optimization. *J. Mol. Biol.*, 236:918–939, 1994.

[100] R. Lee. Protein model building using structure homology. *Nature*, 356:543–544, 1992.

[101] Z. Li, X. Zhang, and L. Chen. Unique optimal foldings of proteins on a triangular lattice. *Appl. Bioinformatics*, 4(2):105–16, 2005.

[102] Y. Liou, A. Tocilj, P. Davies, and Z. Jia. Mimicry of ice structure by surface hydroxyls and water of a beta-helix antifreeze protein. *Nature*, 406:322–324, 2000.

[103] D. Lipman and W. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.

[104] L. Looger, M. Dwyer, J. Smith, and H. Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature*, 423:185–190, 2003.

[105] K. Lundstrom. Structural genomics on membrane proteins. *Cellular and molecular life sciences*, 63:2597–2607, 2006.

[106] A. Machalek. From genes to proteins: Nigms catalogs the shapes of life, 2001.

[107] S. Malakauskas and S. Mayo. Design, structure and stability of a hyperthermophilic protein variant. *Nat. Struct. Biol.*, 5:470–475, 1998.

[108] M. Marti-Renom, A. Stuart, A. Fiser, R. S. R, F. Melo, and A. Sali. Comparative protein structure modeling of genes and genomes. *Annu Rev Biophys Biomol Struct*, 29:291–325, 2000.

[109] J. M. Mason and K. M. Arndt. Coiled coil domains: Stability, specificity, and biological implications. *ChemBioChem*, 5(2):170–176, 2004.

[110] J. McCammon and S. Harvey. *Dynamics of proteins and nucleic acids.* Cambridge University Press, 1987.

[111] J. McCammon and S. Harvey. Dynamic of proteins and nucleic acids. *Cambridge U.P., New York*, 1989.

[112] C. Mead, J. Manuch, X. Huang, B. Bhattacharyya, L. Stacho, and A. Gupta. Investigating lattice structure for inverse protein folding. *FEBS Journal*, 272:4739–380, 2005.

[113] J. Mendes, R. Guerois, and L. Serrano. Energy estimation in protein design. *Curr. Opin. Struct. Biol.*, 12:441–446, 2002.

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

[115] C. Micheletti, F. Seno, A. Maritan, and J. R. Banavar. Protein design in a lattice model of hydrophobic and polar amino acids. *Physical Review Letters*, 80:10:2237–2240, 1998.

[116] D. Minor and P. Kim. Context is a major determinant of beta-sheet propensity. *Nature*, 371:264–267, 1994.

[117] D. Minor and P. Kim. Context-dependent secondary structure formation of a designed protein sequence. *Nature*, 380:730–734, 1996.

[118] A. Morris, M. MacArthur, E. Hutchinson, and J. Thornton. Stereochemical quality of protein structure coordinates. *Proteins: Structure, Function and Genetics*, 12:345–364, 1992.

[119] S. Mosimann, R. Meleshko, and N. James. A critical assessment of comparative molecular modeling of tertiary structures of proteins. *Proteins: Structure, Function and Genetics*, 23:301–317, 1995.

[120] A. Murzin, A. Lesk, and C. Chothia. Principles determining the structure of beta-sheet barrels in proteins. i. a theoretical analysis. *Journal of Molecular Biology*, 236(5):1369–1381, 1994.

[121] N. Naganoa, M. Otaa, and K. Nishikawa. Strong hydrophobic nature of cysteine residues in proteins. *FEBS Letters*, 458(8):69–71, 1999.

[122] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[123] NIAS DNA Bank. Growth of weekly update of protein sequence databases. http://www.dna.affrc.go.jp/growth/P-daily.html, 2008.

[124] N. L. Ogihara, G. Ghirlanda, J. W. Bryson, M. Gingery, W. F. DeGrado, and D. Eisenberg. Design of three-dimensional domain-swapped dimers and fibrous oligomers. *Proc Natl Acad Sci USA*, 98(4):1404–1409, 2001.

[125] J. Onuchic, P. Wolynes, Z. Luthey-Schulten, and N. Socci. Towards an outline of the topography of a realistic protein folding funnel. In *Proc. Natl. Acad. Sci.*, volume 92, pages (8):3626–3630, 1995.

[126] J. Overington. Comparison of three-dimensional structures of homologous proteins. *Current Opinion in Structural Biology*, 2:394–401, 1992.

[127] S. Park, X. Yang, and J. Saven. Advances in computational protein design. *Curr. Opin. Struct. Biol.*, 14:487–497, 2004.

[128] I. Pastan, L. Pai, U. Brinkmann, and D. Fitzgerald. Recombinant toxins: new therapeutic agents for cancer. *Ann. NY Acad. Sci.*, 758:345–354, 1995.

[129] J. Pedersen and J. Moult. Genetic algorithms for protein structure prediction. *Curr. Opin. Struct. Biol.*, 6:227–231, 1996.

[130] M. Petukhov, V. Munoz, N. Yumoto, S. Yoshikawa, and L. Serrano. Position dependence of non-polar amino acid intrinsic helical propensities. *Journal of Molecular Biology*, 278:279–289, 1998.

[131] N. Pierce, J. Spriet, J. Desmet, and S. Mayo. Conformational splitting: A more powerful criterion for dead-end elimination. *J. Comp. Chem.*, 21:999–1009, 2000.

[132] N. A. Pierce and E. Winfree. Protein design is np-hard. *Protein Engineering*, 15:779–782, 2002.

[133] N. Pokala and T. Handel. Review: protein designwhere we were, where we are, where were going. *J. Struct. Biol.*, 134:269–281, 2001.

[134] J. Ponder and F. Richards. Tertiary templates for proteins. use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of Molecular Biology*, 193:775–791, 1987.

[135] J. Ponder and F. Richards. Use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of Molecular Biology*, 193:775–791, 1987.

[136] Protein Data Bank. http://www.wwpdb.org/index.html.

[137] T. Quinn, N. Tweedy, R. Williams, J. Richardson, and D. Richardson. Betadoublet: de novo design, synthesis and characterization of a -sandwich protein. *Proc. Natl Acad. Sci.*, 91:8747–8751, 1994.

[138] D. Ripoll and H. Scheraga. On the multiple minima problem in the conformational analysis of polypetides. *Biopolymers*, 30:165–176, 1990.

[139] G. C. Rodakis and F. C. Kafatos. Origin of evolutionary novelty in proteins: How a high-cysteine chorion protein has evolved. *Proc. Natl. Acad. Sci. USA*, 79:3551–3555, 1982.

[140] F. Salemme. Structural properties of protein beta-sheets. *Prog Biophys Mol Biol.*, 42(2-3):95–133, 1983.

[141] C. Schafmeister and R. Stroud. Helical protein design. *Curr. Opin. Chem. Biotechnol.*, 9(4):350–353, 1998.

[142] H. Schenck and S. Gellman. Use of a designed triple-stranded antiparallel beta-sheet to probe beta-sheet cooperativity in aqueous solution. *J. Am. Chem. Soc.*, 120:4869–4870, 1998.

[143] H. Schrauber, F. Eisenhaber, and O. Argos. Rotamers, to be or not to be? *Journal of Molecular Biology*, 230:592–612, 1993.

[144] E. Shakhnovich. Protein design: a perspective from simple tractable models. *Folding and Design*, 3:R45–R58, 1998.

[145] E. Shakhnovich and A. Gutin. Engineering of stable and fast-folding sequences of model proteins. *Proc. Natl. Acad. Sci.*, 90:7195–7199, 1993.

[146] E. Shakhnovich and A. Gutin. Engineering of stable and fast-folding sequences of model proteins. *Proc. Natl. Acad. Sci.,*, 90:7195–7199, 1993.

[147] K. Simons, R. Bonneau, I. Ruczinski, and D. Baker. Ab initio protein structure prediction of casp iii targets using rosetta. *Proteins: Structure, Function and Genetics*, S3:171–176, 1999.

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

[149] J. Skolnick, A. Kolinski, D. Kihara, M. Betancourt, P. Rotkiewicz, and M. Boniecki. Ab initio protein structure prediction via a combination of threading lattice folding, clustering and structural refinement. *Proteins: Structure, Function and Genetics*, 5:149–156, 2001.

[150] A. Slovic, H. Kono, J. Lear, J. Saven, and W. DeGrado. Computational design of water-soluble analogues of the potassium channel kcsa. *Proc. Natl. Acad. Sci.*, 5:1828–1833, 2004.

[151] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[152] M. Sommerhalter, R. Lieberman, and A. Rosenzweig. X-ray crystallography and biological metal centers: Is seeing believing? *Inorg. Chem*, 44:770–778, 2005.

[153] A. Street and S. Mayo. Computational protein design. *Structure*, 7:R105–R109, 1999.

[154] N. Summers and M. Karplus. Modeling of side chains, loops and insertions in proteins. *Meth. Enzym*, 202:156–205, 1991.

[155] S. Sun, R. Brem, H. Chan, and K. Dill. Designing amino acid sequences to fold with good hydrophobic cores. *Protein Engineering*, 8(l2):1205–1213, 1995.

[156] M. Sundstrom, M. Norin, and A. Edwards. *Structural genomics and high throughput structural biology*. CRC Press, 2005.

[157] M. Swindells and J. Thronton. Modeling by homology. *Current opinion in structural biology*, 290:757–779, 1999.

[158] G. Tuchscherer, L. Scheibler, P. Dumy, and M. Mutter. Protein design: on the threshold of functional properties. *Biopolymers*, 47(1):63–73, 1998.

[159] W. van Gunsteren. Validation of molecular dynamics simulation. *Journal of Computational Physics*, 108:6109–6116, 1998.

[160] G. Veriend. Whatif: a molecular modeling and drug design program. *Journal of Molecular Graphics*, 8:52–56, 1990.

[161] C. Voigt, D. Gordon, and S. Mayo. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.*, 299:789–803, 2000.

[162] R. Weaver. *Molecular Biology*. McGraw-Hill Companies, Inc., 2001.

[163] J. Xu. Protein structure prediction by linear programming, phd thesis, 2003.

[164] Y. Xu, D. Xu, and J. Liang. *Computational methods for protein structure prediction and modeling*. Springer, 2007.

[165] Y. B. Yu. Coiled-coils: stability, specificity, and drug delivery potential. *Advanced Drug Delivery Reviews*, 54(8):1113–1129, 2002.

[166] K. Yue and K. Dill. Inverse protein folding problem: Designing polymer sequences. *Proc. Natl. Acad. Sci. USA*, 89:4163–4167, 1992.

[167] K. Yue, K. Fiebig, P. Thomas, H. Chan, E. Shakhnovich, and K. Dill. A test of lattice protein folding algorithms. *Proc Natl Acad Sci USA.*, 92(1):325–329, 1995.