Computational Methods for Lattice Protein Models

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat
der Technischen Fakulät
der Albert-Ludwigs-Universität Freiburg

2011

von Diplom-Bioinformatiker (Univ.)

Martin Mann

Dekan:

Prof. Dr. Bernd Becker

Prüfungskommission:

Prof. Dr. Christoph Scholl - Vorsitz

Prof. Dr. Jan Korvink - Beisitz

Prof. Dr. Rolf Backofen - Gutachter

Prof. Dr. Ivo L. Hofacker - Gutachter

Datum der Promotion: 27.06.2011

Danke . . .

. . . dass das alles möglich war!

Diese Arbeit wäre nie entstanden ohne die direkte oder indirekte Mitwirkung verschiedenster Menschen, die mich auf meinem Weg begleitet haben. All diesen sind die folgenden zwei Seiten gewidment, damit dies nie in Vergessenheit gerät.

Da ist ganz wichtig und vorne weg natürlich Prof. Rolf Backofen, der durch hervorragende akademische Rahmenbedingungen mein Forschen und Wirken erst ermöglichte und mir mit Rat und Tat zur Seite stand.

Dennoch hätte ich wohl nicht solch eine Fülle an interessanten Projekten bewältigt, wenn ich dabei allein gewesen wäre. Ein Team und gelungene Arbeitsteilung ergibt mehr als die Summe seiner Teile und so bin ich sehr dankbar mit so vielen fähigen und unterhaltsamen Menschen zusammengearbeitet zu haben. Denn Malochen ohne Spaß macht hirntot.

Zudem gilt mein Dank Prof. Ivo Hofacker, der sich die Zeit nimmt mein Niedergeschriebenes zu begutachten, und all den fleissigen Helfern die hier und dort zu seinem Entstehen und zur Güte beigetragen haben.

I want to say THANK YOU to the whole Bioinformatics group including former members and my colleagues all over Europe. You've been a bit like family in the last years.

Und meinen Freunden hier und überall sei gedankt mich stetig daran zu erinnern, dass das Leben aus mehr als deadlines, papern, coden und knows the vulture besteht. Dass man ab und an auch mal raus muss und das Leben genießen.

Last but not least, meine Familie. Ein Zentrum zum Ruhen, ein Quell der Freude, ein Grund zum Verreisen und das Ziel zum Erstreben. Nichts wäre ohne ihre Unterstützung und Liebe. Doch aufgestockt wird alles durch Anke, die mich so lange schon erträgt mit all meinen Schrullen und mir ob nah oder fern doch tief im Inneren wohnt.

Martin Mann

Thanks to ...

Alessandro Dal Palù, Alexander Ullrich, Andreas Richter, Anke Busch (thanks for all and even more), Bela Bender, the bike group of the faculty (I would not know so cool trails), Cameron Smith, Charlotte Deane, Christoph Flamm (next thesis in chemistry?), Daniel Erny, Daniel Maticzka, Deutsche Bahn (ICE rulez!), Dominic Rose, Dragos Alexandru Sorescu, Evandro Ferrada, Fabrizio Costa, Guido Tack, Hannes Kochniß (my brother in heart), Heinz Ekker, Ivo L. Hofacker, Janice Kielbassa (two brains, one fit of laughter), Kathleen Steinhöfel, Klemens Hertel, Konstantin Klemm, Kousik Kundu, Laura Raden (the best Mampfkumpan ever), Lisa Irion, Mathias Möhl, Matthias Brosemann (the god of coding), Michael “Hillerich” Hiller (have still not reached the 10 papers a year), Michael “MTW” Wolfinger, Mohamad Rabbath, Monica “Mona” Möbes, Monika Degen-Hellmuth (the heart and soul of the group), Mone Kraft and the magnificient Bingo Bär, my bachelor, master, and course students (I better understand when I’ve got to explain it), Omer Alkhnbashi, Rhodri Saunders (BIG LOVE), Rileen Sinha, Robert Kleinkauf, Rolf Backofen (always ready to be the best), Peter Stadler, Sander Ziegenrücker (no better friend all around the world), Sebastian Will (no way to be so far without your help), Sita Lange (the fresh breeze with giggle and love), Stefan Jankowski (there is no technical detail he cannot explain), Steffen Heyne, Steffi Pfau (Elite-WGlerin vorm Herrn)

...for my great time in Freiburg!

TABLE OF CONTENTS

ABSTRACT

Proteins are involved in almost all processes in living cells. They act as regulators, catalyzers, transporters, and in many other functions that are determined by their three-dimensional structures.

This thesis studies the fundamental concepts that define and guide the folding processes of proteins. Therein, the prediction of a protein's native fold as well as the modeling of its folding process are of great importance. To enable large scale studies, lattice protein models are used that are available at different levels of abstraction. Central to this thesis is the development and implementation of efficient methods to study proteins represented in complex three-dimensional lattices. A major focus is the development of procedures that enable the usage of more realistic side chain models.

An important task when studying protein models is the transfer of real protein structures into the model. This NP-complete problem is tackled in the first part of the thesis. A combination of efficient heuristics and constraint-based search yields models of high quality and low runtimes.

The second part of the thesis presents methods to determine minimum energy structures. Here, a constraint-based approach is introduced that for the first time makes it possible to predict energetically optimal structures within hydrophobic-polar (HP) side chain models. This enables the first study of optimal structures within the model revealing an immense degeneracy. Since many structures cannot be distinguished by the energy function, an equivalence relation for the grouping of optimal structures is introduced. An extension of the constraint-based structure prediction approach enables the efficient and direct computation of the resulting equivalence classes. HP-optimal structures from different classes can be used for the initialization of local search methods that tackle more advanced energy functions. The superiority of such an approach compared to standard strategies is demonstrated.

In addition, to enable local search methods in side chain models, the definition and efficient implementation of a neighborhood relation between structures is needed. The third part of the thesis covers the presentation of an interval-based local neighborhood relation for arbitrary lattices. An efficient procedure for the enumeration of neighbored structures opens the door for new studies in side chain lattice protein models.

Subsequently, the phenomenon of co-translational folding is explored, i.e. the folding of the emerging protein while it is assembled at the ribosome. Co-translational folding is assumed to guide the folding process into the native structure. The introduced methods enable a classification of protein sequences based on their co-translational folding potential. An extensive, comparative study identifies new characteristics in sequence and structure that are exclusive to co-translationally folding proteins. Furthermore, some hypotheses from literature are disproved that have been proposed based on thought experiments. An extension of the study to real protein structures and domains highlights the α/β-domain proteins. This class shows the strongest bias towards the identified characteristics of co-translational folding proteins.

In the final part of the thesis the focus is shifted to evolutionary studies. Therein, intensive analyses of neutral networks are done that are graph-based tools to study neutral evolution. Neutral networks describe the possible evolutionary pathways that preserve a given function and thus the associated structure. A new sequence design approach is introduced that enables the neutral network exploration without a full sequence space enumeration. This is the first method that is able to design non-degenerated sequences for a given structure, which is known to be a difficult, NP-complete problem. A thorough analysis of the resulting neutral networks in three-dimensional lattice models reveals considerable differences, e.g. in network sizes, compared to two-dimensional models. To focus the investigation of neutral evolution on the structural core of proteins an according H-fold definition is presented. The H-folds enable additional evolutionary studies of the flexible loop regions of proteins.

In conclusion, this thesis describes a variety of new and efficient methods that enable extensive studies of structures and sequences in lattice protein models. All methods are freely available for further research within two software packages and via a web frontend for ad hoc usage. The implemented tools as well as the studies presented thus provide an important contribution to *in silico* protein research.

ZUSAMMENFASSUNG

In fast alle lebenswichtigen Prozesse in lebenden Zellen sind Proteine involviert. Sie wirken hierbei als Regulatoren, Katalysatoren, Transporter und in einer Vielzahl von anderen Funktionen, die durch ihre dreidimensionale Struktur bestimmt werden.

Diese Arbeit befasst sich mit den grundlegenden Zusammenhängen, die der Strukturbildung von Proteinen zugrunde liegen. Dabei ist die Vorhersage ihrer funktionalen Form von zentraler Bedeutung sowie die Modellierung des Faltungsprozesses an sich. Um möglichst umfangreiche Studien zu ermöglichen, werden hierzu Gitterproteinmodelle verwendet, die in einer Vielzahl von Abstraktionsgraden verfügbar sind. Der Fokus der Arbeit liegt auf der Entwicklung und Umsetzung von effizienten Methoden zum Studium von Gitterproteinen in komplexen dreidimensionalen Gittern. Hierbei stehen insbesondere Verfahren im Vordergrund, welche die Verwendung von Seitenkettenmodellen ermöglichen.

Ein wichtiger Schritt ist hierbei der Transfer von realen Proteinstrukturen in das verwendete Modell. Dieses NP-vollständige Problem wird im ersten Teil der Arbeit behandelt. Eine Kombination aus effizienten Heuristiken und constraint-basierter Suche liefert Modelle von sehr hoher Qualität in kurzer Laufzeit.

Der zweite Teil der Arbeit geht auf Methoden zur Bestimmung von energieminimalen Strukturen ein. Auch hier kommt ein constraint-basierter Ansatz zum Einsatz der erstmals die Bestimmung von energieoptimalen Strukturen im HP-Seitenkettenmodell ermöglicht. Daraufhin erfolgt die erste Studie optimaler Konformationen in diesem Modell, wobei eine enorme Degeneriertheit beobachtet wird. Da viele Strukturen mit der verwendeten HP-Energiefunktion nicht unterscheidbar sind, wird eine entsprechende Äquivalenzrelation zur Gruppierung ähnlicher Strukturen eingeführt. Eine Erweiterung des constraint-basierten Ansatzes zur Strukturbestimmung ermöglicht die effiziente und direkte Aufzählung der resultierenden Äquivalenzklassen. HP-optimale Strukturen aus verschiedenen Klassen können zur Initialisierung von lokalen Suchverfahren unter Verwendung von umfassenderen Energiemodellen verwendet werden. Die Überlegenheit eines derartigen Ansatzes gegenüber herkömmlichen Strategien wird demonstriert.

Um lokale Suchverfahren auch in Seitenkettenmodellen zu ermöglichen, ist die Definition und effiziente Umsetzung einer Nachbarschaft von Strukturen notwendig. Eine intervallbasierte lokale Nachbarschaftsrelation sowie ein effizientes Verfahrung zu seiner Aufzählung umfasst der dritte Teil der Arbeit.

Anschließend wird das Phänomen der cotranslationalen Faltung untersucht. Dieses beschreibt den Vorgang der Strukturbildung noch während der Konstruktion eines Proteins am Ribosom. Es wird angenommen, dass hierdurch die Faltung in die funktionelle Konformation begünstigt wird. Die eingeführten Verfahren ermöglichen eine Klassifikation von Proteinsequenzen anhand ihres cotranslationellen Faltungspotentials. Eine umfangreiche vergleichende Studie identifiziert neue Charakteristika in Sequenz und Struktur welche cotranslational faltenden Proteinen zugeordnet werden können. Zudem werden einige Hypothesen aus der Literatur widerlegt, welche großteilig auf Gedankenexperimenten basierten. Eine Ausweitung der Studie auf reelle Proteinstrukturen und -domänen hebt die Gruppe der α/β-Domänproteine hervor, die die stärksten Trends zu den identifizierten cotranslationalen Charakteristika aufweisen.

Im finalen Teil der Arbeit wechselt der Fokus auf evolutionäre Fragestellungen. Hierbei erfolgt eine umfangreiche Studie von neutralen Netzwerken, einem graphenbasiertem Werkzeug um neutrale Evolution zu untersuchen. Neutrale Netzwerke beschreiben hierbei mögliche evolutionäre Pfade welche eine gegebene Funktion und somit Struktur erhalten. Es wird ein Sequenzdesignverfahren eingeführt, um die Generierung von neutralen Netzwerken zu ermöglichen. Dieses Verfahren ist das erste, welches nicht-degenerierte Sequenzen für eine gegebene Struktur designen kann, einem schweren, NP-vollständigen Problem. Eine umfangreiche Analyse der resultierenden neutralen Netzwerke in dreidimensionalen Gittern zeigt deutliche Unterschiede, u.a. in ihrer Größe, zu zweidimensionalen Modellen auf. Um die Untersuchung von neutraler Evolution auf den strukturellen Kern von Proteinen zu fokussieren, wird eine neue Definition von sogenannten H-folds eingeführt. Diese ermöglicht darüberhinaus Studien über die Evolution der flexiblen Teile von Proteinen, welche die höchste Sequenzvariation aufweisen.

Zusammenfassend beschreibt diese Arbeit eine Vielzahl von neuen, effizienten Verfahren, welche umfangreiche Studien von Strukturen und Sequenzen in Gitterproteinmodellen ermöglichen. Alle Methoden stehen in zwei Softwarepaketen und durch ein entsprechendes web-frontend frei für weitere Forschung zur Verfügung. Die entstandenen Programme sowie die mit ihnen ermöglichten Studien liefern somit einen wichtigen Beitrag zur computergestützten Proteinforschung.

PUBLICATIONS

Publications covered by this Thesis

- **Signatures of Co-translational Folding**
  R. Saunders**,* ***M. Mann*****, and C. Deane*
  in *Biotechnology Journal, Special Issue on Protein Folding in vivo*, 2011, vol. 6(6).
  * These authors contributed equally to this work.
  doi:10.1002/biot.201000330

- **Lattice model refinement of protein structures**
  M. Mann *and A. Dal Palù*
  in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'10)*, 2010
  arXiv:1005.1853

- **Protein Folding Simulation by Two-stage Optimization**
  A. D. Ullah, L. Kapsokalivas, ***M. Mann*** *and K. Steinhöfel*
  in *Proc. of the International Symposium on Intelligence Computation and Applications (ISICA'09), CCIS*, vol. 51, pp. 138-145, Springer, 2009
  doi:10.1007/978-3-642-04962-0_16

- **CPSP-web-tools: a server for 3D lattice protein studies**
  M. Mann*, C. Smith, M. Rabbath, M. Edwards, S. Will, and R. Backofen*
  in *Bioinformatics*, vol. 25 (5), pp. 676–677, 2009
  doi:10.1093/bioinformatics/btp034

- **Constraint-based Local Move Definitions for Lattice Protein Models Including Side Chains**
  M. Mann*, A. Hamra, K. Steinhöfel, and R. Backofen*
  in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'09)*, 2009
  arXiv:0910.3880

- **Equivalence Classes of Optimal Structures in HP Protein Models Including Side Chains**
  M. Mann*, R. Backofen and S. Will*
  in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'09)*, 2009
  arXiv:0910.3848

- **CPSP-tools - exact and complete algorithms for high-throughput 3D lattice protein studies**
  M. Mann*, S. Will, and R. Backofen*
  in *BMC Bioinformatics*, vol. 9, pp. 230, 2008
  doi:10.1186/1471-2105-9-230

- **Classifying protein-like sequences in arbitrary lattice protein models using LatPack**
  M. Mann*, D. Maticzka, R. Saunders, and R. Backofen*
  in *HFSP Journal, Special Issue on Protein Folding: Experimental and Theoretical Approaches*, vol.

2(6), pp. 396, 2008
doi:10.2976/1.3027681, PMID: 19436498, PMCID: PMC2645588

- **Decomposition during search for propagation-based constraint solvers**
  M. Mann*, G. Tack, and S. Will*
  Technical report, ALU Freiburg, 2007
  arXiv:0712.2389

- **Counting protein structures by DFS with dynamic decomposition**
  S. Will and ***M. Mann***
  in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'06)*, pp. 83–90, 2006

Further Publications

- **Efficient exploration of discrete energy landscapes**
  M. Mann *and K. Klemm*
  in *Physical Review E*, vol. 83(1), pp. online, 2011
  doi:10.1103/PhysRevE.83.011113

- **Shape-based barrier estimation for RNAs**
  S. Bogomolov,* ***M. Mann****, B. Voss, A. Podelski, and R. Backofen*
  in *Proc. of German Conference on Bioinformatics (GCB'10)*, vol. 173 of LNI, pp. 42-51, GI, 2010
  * These authors contributed equally to this work.

- **Evolution of metabolic networks: A computational framework**
  C. Flamm, A. Ullrich, H. Ekker, ***M. Mann****, D. Hoegerl, M. Rohrschneider, S. Sauer, G. Scheuermann, K. Klemm, I. L. Hofacker, P. F. Stadler*
  in *Journal of Systems Chemistry*, vol. 1(1), pp. 4, 2010
  doi:10.1103/PhysRevE.83.011113

- **Techniques de dcomposition pour lisomorphisme de sous-graphe**
  (Decomposition techniques for subgraph matching)
  S. Zampelli, ***M. Mann****, Y. Deville, and R. Backofen*
  in *Proc. of the 4th Journes Francophones de Programmation par Contraintes (JFPC'08)*, 2008
  arXiv:0805.1030 (english version)

- **The energy landscape library - a platform for generic algorithms**
  M. Mann*, S. Will, and R. Backofen*
  in *Proc. of the 1st international Conference on Bioinformatics Research and Development (BIRD'07)*, vol. 217, pp. 83-86, OCG, 2007

CHAPTER I

Introduction

1.1 Proteins and Protein Models

Proteins are the central players in the game of life. Involved in almost all cellular processes, they are essential for any life-form on earth (*Tyers and Mann*, 2003).

It is central to biology that proteins are encoded by genes of the DNA, the genetic memory of the cell. Such genes are templates for transcription to create messenger RNA (mRNA) molecules, the working copies of genes. During translation, these mRNAs are decoded into proteins, the workhorses of life.

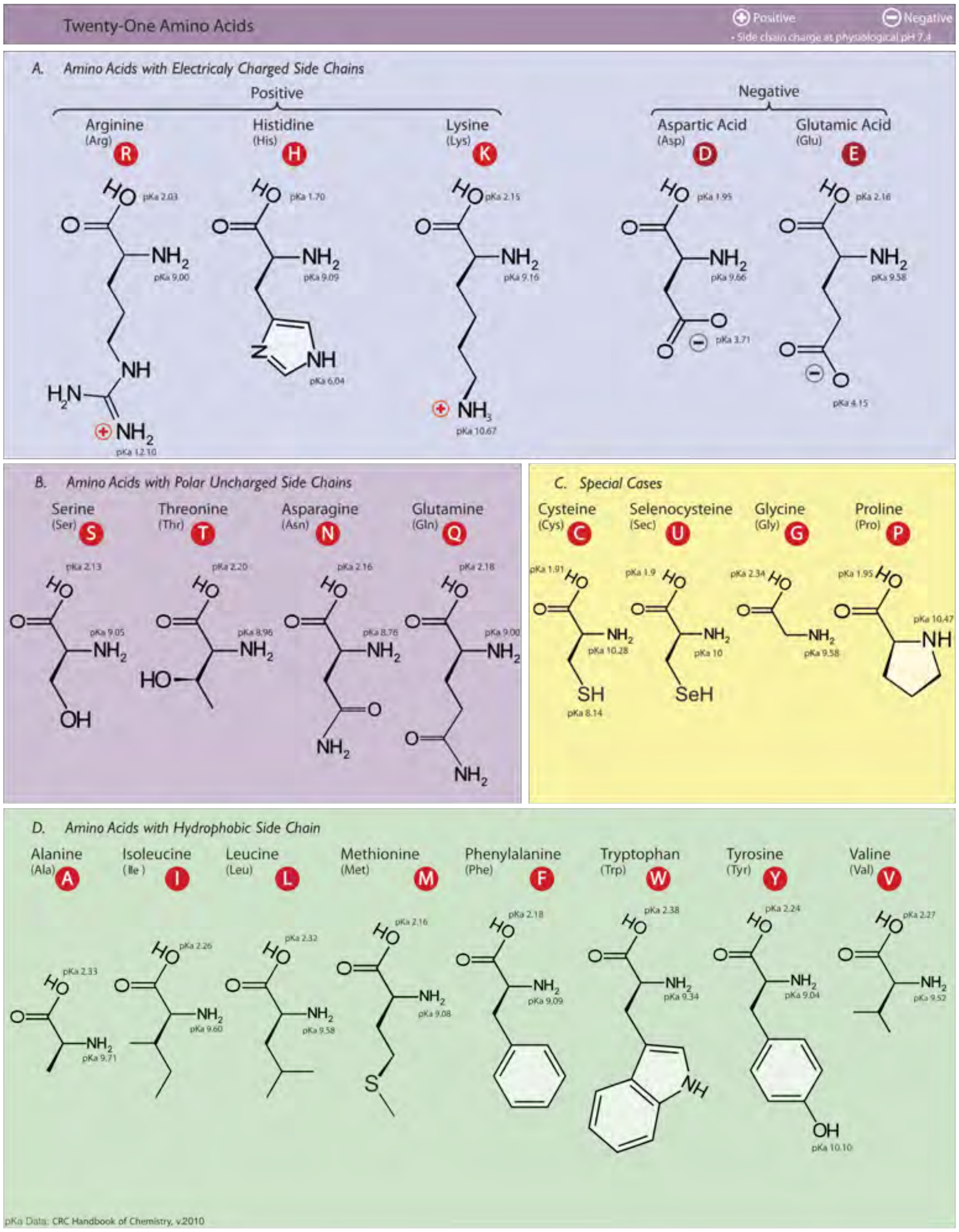
After translation, each protein has to adapt its specialized structure that is needed to perform its function. In the following this functional structure is referred to as the *functional, native, or biological fold/conformation. In vitro* refolding experiments have shown that the protein's functional fold is mainly encoded by its sequence of amino acids, i.e. its *primary structure* (*Anfinsen*, 1973; *Crippen*, 1991). Nevertheless, there is a large number of processes influencing or supporting the structure formation process within living cells (*Clark*, 2004). Among them are crowding effects (*Jefferys et al.*, 2010), the presence of helper molecules like chaperons (*Ying et al.*, 2006), the speed of translation (*Thanaraj and Argos*, 1996) that influences the effect of co-translational folding (*Kramer et al.*, 2009; *Ellis et al.*, 2010), and the binding of special co-factors (*Wittung-Stafshede*, 2002), to name only a few.

Still, the knowledge of a proteins primary structure or sequence is essential to investigate its structure mediated function. Proteins are chain-like polymers of amino acids. Figure 1.1[1] depicts the most common *proteinogenic* amino acids that are found in proteins. All of them show a common pattern, i.e. a backbone formed by an amine group, a carbon atom (C_α), and a carboxyl group as well as a varying side chain attached to the C_α-atom (see Fig. 1.2[2]). Thus, amino acids only differ in their side chains (or residues) that induce different chemical and physical properties.

During translation amino acids are linked to form the protein. To this end, the carboxyl group of the last amino acid forms a covalent chemical bond with the amine group of the added amino acid. The formed connection is named a *peptide bond* (see Fig. 1.2). Thus, the resulting so called polypeptide or protein exhibits a synthesis mediated direction, i.e. it starts with a free amine group, *the amino- or N-terminus*, and ends with a free carboxyl group, *the carboxy- or C-terminus.*

[1] Figure by Dan Cojocari 2010; available at `http://en.wikipedia.org/wiki/File:Aa.svg`

[2] Figure by Yassine Mrabet 2007; available at `http://en.wikipedia.org/wiki/File:AminoAcidball.svg`



Dan Cojocari, Department of Medical Biophysics, **University of Toronto**, 2010

Figure 1.1: The twenty-one proteinogenic amino acids found in eukaryotes, grouped according to properties of their side chains. Note, the newly discovered 21st amino acid Selenocystein is not found in all organisms (*Bock et al.*, 1991) and will be neglected in the manuscript. Therefore, only twenty amino acids (excluding `U`) are considered.

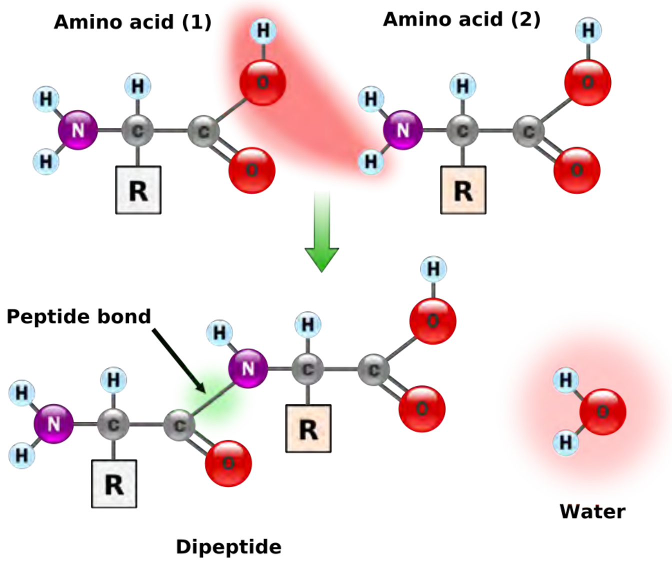


Figure 1.2: The general structure of proteinogenic amino acids is given in the upper part: the left part forms the amine group (NH_2) and the right part the carboxyl group ($COOH$). The R groups represent the amino acid specific side chains, see Fig. 1.1, and are attached to the central carbon atoms (C_α). The whole picture depicts the peptide bond formation between two amino acids where a water molecule is produced.

In cells, the translation process is mediated by ribosomes, large RNA molecules complexed with helper proteins (*Clark*, 2004). The growing protein is extruded into the cell and directly starts to form a structure driven by thermodynamics. This phenomenon is called *co-translational folding* and will be studied extensively in Chapter V. Some proteins need the assistance of helper proteins, namely chaperons, to fold into their functional structure (*Ying et al.*, 2006). Such chaperons prevent misfolding of protein parts or protein aggregation via temporary docking and thereby guide the folding process.

Overall, the folding process is governed by non-covalent intramolecular interactions (*Rose et al.*, 2006). Here, single atoms or groups attract or repulse each other based on their chemical properties. The strongest influence on the folding process is mediated by the hydrophobicity of some amino acids (*Pace et al.*, 1996). They are repelled from water due to their non-polar nature and thus prefer the neighboring of other hydrophobic residues. Thus, the hydrophobic protein parts are herding since the surrounding cellular fluid consists mostly of water. This results in a so called *hydrophobic core* present in almost all globular water-solved protein structures. Local, so called *secondary structure elements*, are formed by hydrogen bonding. *Pauling and Corey* (1951a,b) characterized two major forms: α-helix and β-sheet, where the first appears spiraled and the second is formed by a parallel or anti-parallel alignment of subchains (see Fig. 1.3). Further forces like Van der Waals interactions, electrostatics of charged residues, and disulphide bridges between cysteines are guiding the folding into the native structure. The three-dimensional (3D) fold of a single sequence is called the *tertiary structure* while a further conglomeration of several protein chains into one multi-subunit complex is named a *quaternary structure*. Within this manuscript, only single chain structures are studied.

The final, native structure of the protein defines it function. If misfolding occurs then the protein's functional capabilities are at least restrained and several diseases are thus connected to

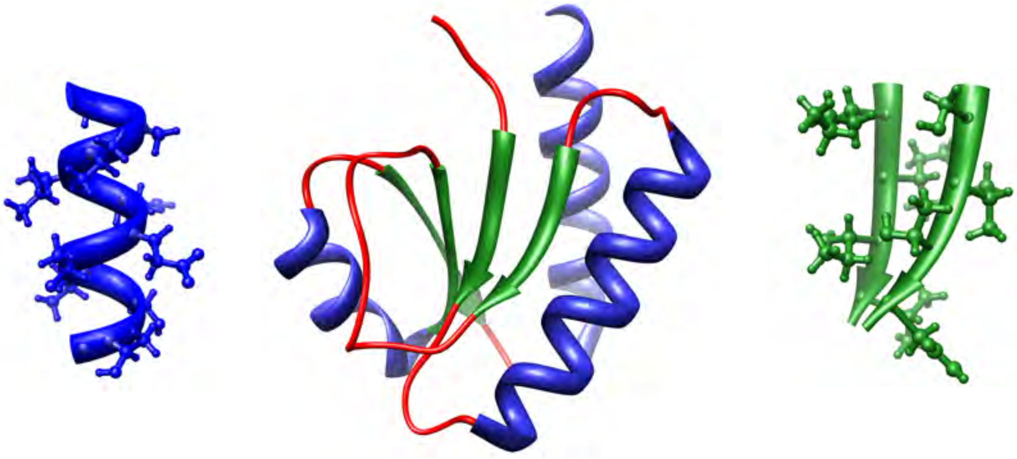


Figure 1.3: Secondary structure elements in protein structures: (left) **α-helix**, (right) **β-sheet**, and flexible **loop structures**. The middle shows the *tertiary structure* of a protein domain (PDB ID `1VKR`) with highlighted secondary structure elements. The flanking figures depict the side chain configuration in α-helix and β-sheet structures. Note, within α-helical structures, all side chains point to the outside where as in β-sheet structures the side chains alternate between an outwards and inwards orientation relative to the plane of the imaginary sheet.

misfolding problems (*Smith*, 2003). Well known examples are the bovine spongiform encephalopathy (BSE, also known as "mad cow disease") in cattle and Creutzfeldt-Jakob disease (CJD) in human caused by the misfolding of prions (*Prusiner*, 1998; *Nunnally and Krull*, 2003; *Laurèn et al.*, 2009). Aberrant folding is also associated to cancer (*Scott and Frydman*, 2003).

The study of proteins and their folding processes enables new perspectives for medical treatment (*Alekshun and Levy*, 2007), drug design (*Simmons et al.*, 2010), and diagnostics (*Theurillat et al.*, 2010). Beyond that it facilitates new bio-technical innovations, since proteins are fundamental in all metabolisms. Examples are the use of specialized bacteria for the degradation of wastewater (*Shannon et al.*, 2008), sludge (*Rivière et al.*, 2009), or oil (*Nature-Editorial*, 2010).

For large scale studies of proteins and their folding processes computational simulations and models are needed (*Ding and Dokholyan*, 2002). Depending on the application and the focus of interest a careful selection of the level of abstraction has to be done to find a trade-off between simplicity (computational complexity) and expressivity (the relevance of the results). Therefore, a large variety of protein models has been introduced in the last decades. When comparing models, one can distinguish between the abstraction of:

1. Structure space
2. Sequence space
3. Energy function

The first abstraction restricts the set of possible (tertiary) structures a protein can adopt. This ranges from full atom representations in three dimensions for molecular dynamics simulations (*Karplus and Kuriyan*, 2005); through off-lattice bead models (*Angelani and Ruocco*, 2009); and down to very coarse grained 2D lattice models (*Lau and Dill*, 1989). Figure 1.4 exemplifies

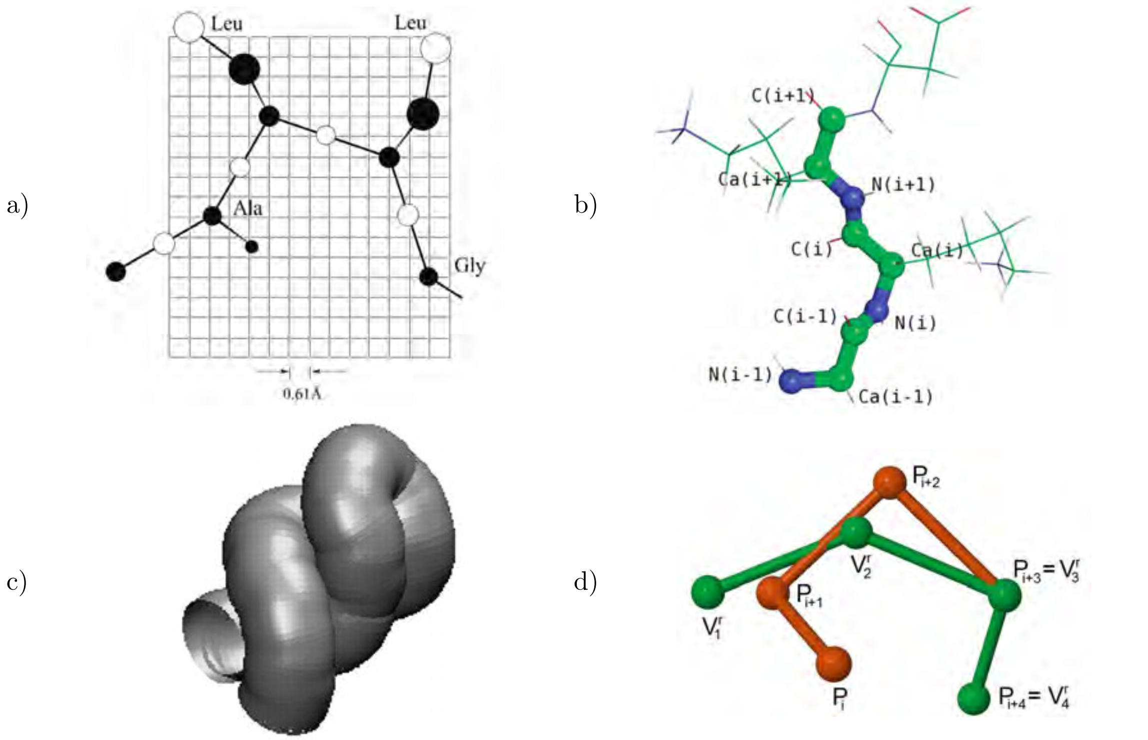


Figure 1.4: Examples of different protein models used in literature: (a) the CABS model (taken from (*Ekonomiuk et al.*, 2005)), (b) the backbone abstraction (in bold) used by *Ponty et al.* (2008) (both (a/b) are explained in Sec. 1.2.2.4), (c) the tube model introduced by *Hoang et al.* (2004) (taken from (*Banavar et al.*, 2007)), and (d) the 4-tuple fragment-based discretized off-lattice protein model by *Dal Palù et al.* (2010).

some protein models used in literature. In general, two types of structural abstraction can be distinguished: the simplification of real 3D coordinates to discretized coordinates and the consolidated representation of all atoms of an amino acids by a single sphere. Both is done within lattice protein models that are studied within this thesis.

The sequence space describes the model of the amino acid (primary) sequences represented within the protein model defined by the alphabet sequences are based on. Alphabets range from the intuitive set of all 20 proteinogenic amino acids as given in Fig. 1.1 to the abstract representation of certain features of each amino acid, e.g. its hydrophobicity/polarity within the HP model (*Lau and Dill*, 1989) that will be discussed in detail within this thesis.

The abstraction of structure and sequence space are mainly independent from one another, but both restrain the abstraction of the energy function model. The energy function covers the inter- and intra-molecular forces that guide the protein folding process discussed above. The modeled *Gibbs (free) energy* for a given structure measures (simply spoken) the amount of work that is needed to form the structure (*Atwood et al.*, 2006). The lower the energy value of a structure the more favorable it is.

Following the assumption that a protein reaches its thermodynamic equilibrium during folding (*Finkelstein and Badretdinov*, 1997), the native structure will be the structure with minimal free energy (*Anfinsen*, 1973). Therefore, protein folding can be simulated as an energy minimization process based on a given energy function. Furthermore, we might directly solve this energy minimization problem and thus be able to calculate the native (minimal energy) structure.

The most realistic energy functions for full atom protein models try to incorporate all knowledge about folding forces to reflect the biological process as closely as possible (*Alder and Wainwright*, 1959). This comes at the cost of immense computational complexity and limits the application (*Plimpton*, 1995). Thus, depending on the represented structure and sequence space, the energy function abstracts as well. Often distance based potentials, as given by *Tobi and Elber* (2000), are used in combination with steric contributions, the latter e.g. for preferred backbone angles etc. Most simplifying are contact potentials that only consider very close interactions (*Miyazawa and Jernigan*, 1996).

The methods presented within this thesis consider lattice protein models only. Therefore, these models are introduced in detail within the next section.

1.2 Lattice Protein Models

Lattice protein models are a common abstraction of proteins as illustrated in the recent review by *Istrail and Lam* (2009). They are used to investigate the folding process (*Potzsch et al.*, 2006), native structure properties (*Hart and Newman*, 2006), sequence evolution (*Bornberg-Bauer*, 2002), cooperative/competitive folding (*Citossi and Guigliarelli*, 2005), and co-translational folding (*Huard et al.*, 2006), to name but a few.

The applied large-scale studies are only possible using a high level of abstraction; they are infeasible in more realistic full atom protein representations due to the vast size of sequence and structure space. The strongest simplification is done in structure space: the representation of each amino acid is reduced to one or two spheres. In addition, the three-dimensional placement of each sphere is restricted to nodes of an underlying discrete lattice. This results in a finite structure space open to full enumeration for short sequence lengths (*Bromberg and Dill*, 1994; *Shakhnovich and Gutin*, 1993). How precisely a lattice protein structure is able to model a real protein conformation depends strongly on the underlying lattice used (*Park and Levitt*, 1995). Figure 1.5 exemplifies different lattice protein models used in literature.

The energy function is typically contact based, i.e. it sums sequence specific pairwise potentials for amino acids that are in close distance within the conformation. In Fig. 1.5 such contacts are highlighted in red in the 2D-models' drawings. They are typically defined by the minimal distance between neighbored positions within the lattice. More complex energy functions incorporating distance-based potentials and further energy contributions have been used as well (*Sippl*, 1990; *Shen and Sali*, 2006). Usually, the applied energy function directly depends on the modelled sequence space, i.e. the used alphabet. When representing all 20 proteinogenic amino acids from Tab. 1.1, often the pairwise contact potentials suggested by *Miyazawa and Jernigan* (1985, 1996) are used. For reduced alphabets, as the HP model, specific energy functions have been defined (*Lau and Dill*, 1989; *Backofen et al.*, 1999; *Hoque et al.*, 2009).

In the following the required formalisms are introduced to enable a clear presentation of the methods developed in the remaining thesis. In Section 1.2.1 the lattices used within this thesis are defined. This is followed by the most common structure abstractions for lattice protein models (Sec. 1.2.2), namely backbone-only and side chain lattice proteins. Afterwards, different sequence

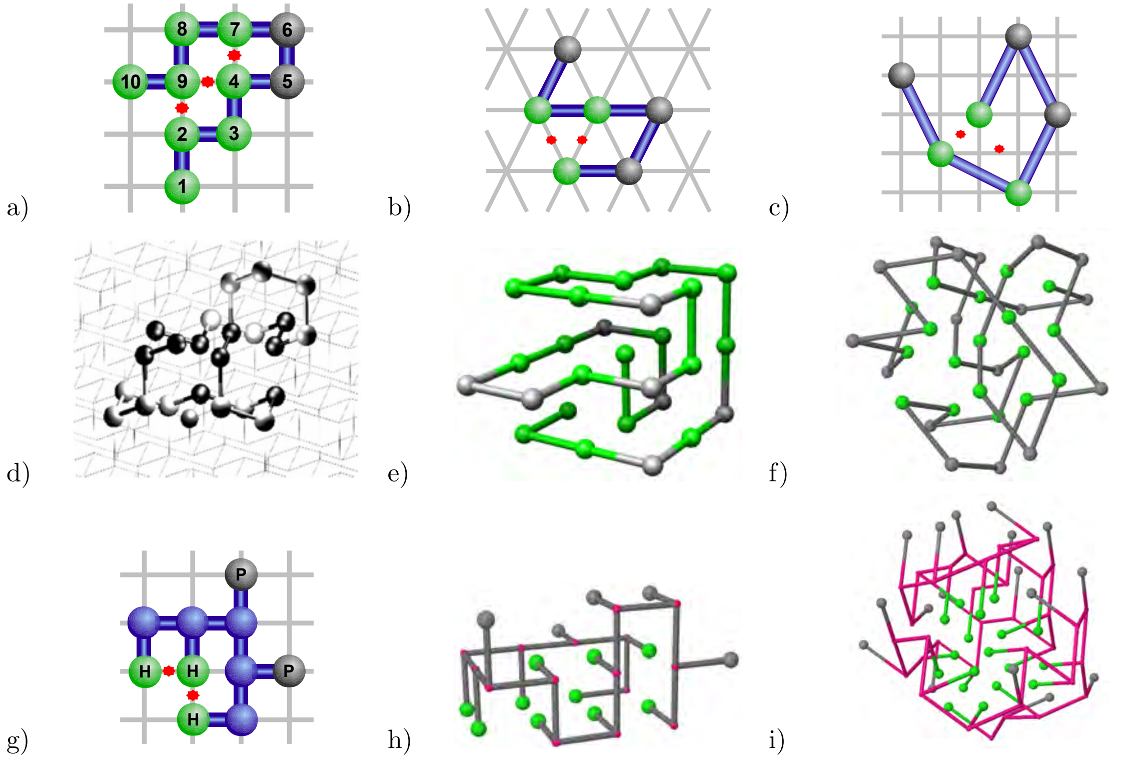


Figure 1.5: Examples of lattice protein models used in the literature: *Backbone-only models* in (a) 2D-square lattice (*Lau and Dill*, 1989), (b) 2D-triangular lattice (*Böckenhauer et al.*, 2008), (c) 2D view of 3D-210 "chess knight" lattice (*Sun et al.*, 1999), (d) 3D-diamond lattice (*Krasnogor et al.*, 2002), (e) 3D-cubic lattice (*Thachuk et al.*, 2007), and (f) 3D-FCC lattice (*Mann et al.*, 2008b). *Side chain models* in (g) 2D-square lattice (*Bromberg and Dill*, 1994), (h) 3D-cubic lattice (*Hart and Istrail*, 1997), and (i) 3D-FCC lattice (*Mann et al.*, 2009c). In 2D models figures (a,b,c,g), favorable contacts are highlighted (red stars).

alphabets and according energy functions are introduced in Sec. 1.2.3.

1.2.1 Lattices

In order to discretize the structure space of proteins, three-dimensional lattices can be used. Such a *lattice* L is a set of 3D coordinates (also named nodes, points, or vectors) that contains the zero coordinate $\vec{0}$ and forms an additive group with the $+$ and $-$ operator for any two points $\vec{u}, \vec{v} \in L$, i.e. it holds that

$$\begin{aligned} L &\subset \mathbb{R}^3 && (1.1) \\ \vec{0} &\in L && (1.2) \\ \vec{u}, \vec{v} &\in L \longrightarrow (\vec{u}+\vec{v}), (\vec{u}-\vec{v}) \in L && (1.3) \end{aligned}$$

For each lattice L, one can identify a minimal set of *neighboring vectors* N_L such all vectors $\vec{u} \in L$ of the lattice can be represented by a linear combination of these vectors using positive integers only, i.e.

$$\forall_{\vec{u} \in L} : \vec{u} = \sum_{\vec{x} \in N_L} c_{\vec{x}} \cdot \vec{x} \quad \text{with } c_{\vec{x}} \in \mathbb{Z}_0^+ \tag{1.4}$$

Furthermore, we enforce that the negation/opposite $-\vec{x}$ of each vector $\vec{x} \in N_L$ of the neighborhood is within the neighborhood as well, i.e.

$$\vec{x} \in N_L \longrightarrow -\vec{x} \in L \tag{1.5}$$

Thus, we say two *nodes* $\vec{u}, \vec{v} \in L$ *are neighbored* within the lattice if their distance vector is a neighborhood vector, i.e.

$$(\vec{u} - \vec{v}) \in N_L \longleftrightarrow \vec{u} \text{ and } \vec{v} \text{ are neighbored.} \tag{1.6}$$

To obtain a *regular lattice*, all vectors of the neighborhood N_L have to be of equal length:

$$\forall_{\vec{x}_1, \vec{x}_2 \in N_L} \; : \; |\vec{x}_1| = |\vec{x}_2| \tag{1.7}$$

where $|\vec{v}|$ denotes the length of the vector $\vec{v}$. This property make the formalized lattices a subgroup of the Bravais lattices where the spanning neighborhood vectors can be of different lengths.

The number of lattice neighborhood vectors $|N_L|$ is an important property of a lattice L and is called its *coordination number*. It is a measure of the lattices complexity.

Table 1.1 lists the lattice neighborhoods of four different lattices that are used within this manuscript. A visualization is given in Fig. 1.6.

1.2.2 Abstraction of Structure Space

Lattice protein models are a coarse abstraction of real proteins where structures are discretized using a lattice. One can distinguish two major classes of lattice protein structures: backbone-only and side chain models.

Within *backbone-only models*, only the protein's backbone is represented with one monomer per

Lattice Name	ID	Neighborhood N_L	$\|N_L\|$
Square	SQR	$\{\pm(1,0,0),\pm(0,1,0)\}$	4
Cubic	CUB	$\{\pm(1,0,0),\pm(0,1,0),\pm(0,0,1)\}$	6
Face Centered Cubic	FCC	$\left\{\begin{array}{c}\pm(1,1,0),\pm(1,0,1),\pm(0,1,1),\\ \pm(1,-1,0),\pm(1,0,-1),\pm(0,1,-1)\end{array}\right\}$	12
Chess Knights Walk	210	$\left\{\begin{array}{c}\pm(2,1,0),\pm(2,-1,0),\pm(2,0,1),\pm(2,0,-1),\\ \pm(1,2,0),\pm(-1,2,0),\pm(1,0,2),\pm(-1,0,2),\\ \pm(0,1,2),\pm(0,-1,2),\pm(0,2,1),\pm(0,2,-1)\end{array}\right\}$	24

Table 1.1: The IDs, neighborhoods N_L, and coordination numbers $|N_L|$ of the lattices used. Note, $\pm\vec{u}$ abbreviates $\vec{u}$ and $-\vec{u}$. See Fig. 1.6 for a visualization.

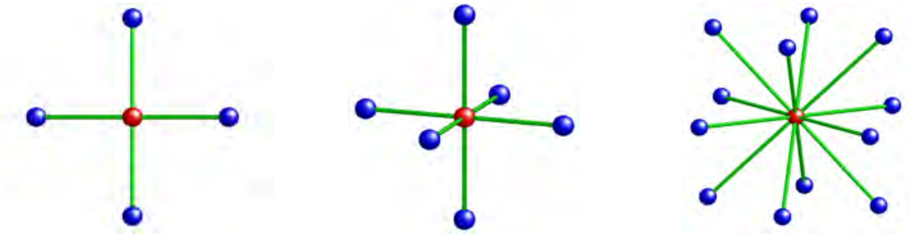


Figure 1.6: Visualization of the lattice neighborhood N_L in different lattices (from left to right: 2D-square, 3D-cubic, and 3D-FCC lattice). The reference point is given in red, the set of neighboring vectors are depicted in green, and the reached neighbored points are colored in blue.

amino acid. In *side chain models*, the abstraction is extended by a second monomer for each amino acid to model its side chain. All monomers are confined to the underlying lattice, such that connected monomers in sequence are neighbored within the lattice. A first graphical sketch is given in Fig. 1.7. In the following, both structure abstractions are discussed in detail.

1.2.2.1 Backbone-only Models

Backbone-only models of proteins usually only represent the C_α-positions of amino acids (*Covell and Jernigan*, 1990). Occasionally their centroids/centers of mass (*Lau and Dill*, 1989) is represented. The representative monomer has no volume nor mass, i.e. is independent of the amino acid side chain size.

Given a lattice L, a *backbone-only lattice protein structure* P of length n is a sequence of lattice nodes $P = (P_1, \ldots, P_n)$ such that all occupied nodes are different and successive monomer nodes are neighbored within the lattice, i.e.

$$\forall_{1\leq i\leq n}: \quad P_i \in L \tag{1.8}$$

$$\forall_{1\leq i<j\leq n}: \quad P_i \neq P_j \qquad \text{i.e. } P \text{ is self-avoiding} \tag{1.9}$$

$$\forall_{1<i\leq n}: \quad (P_i - P_{i-1}) \in N_L \qquad \text{i.e. } P_i \text{ is neighbored to predecessor } P_{i-1} \tag{1.10}$$

Backbone-only model structures in different lattices are exemplified in Fig. 1.5 and on the left in Fig. 1.7. The complexity of the lattice, i.e. its coordination number, directly influences the growth of the resulting structure space. In Table 1.2 and Fig. 1.8 an illustration of structure space size for

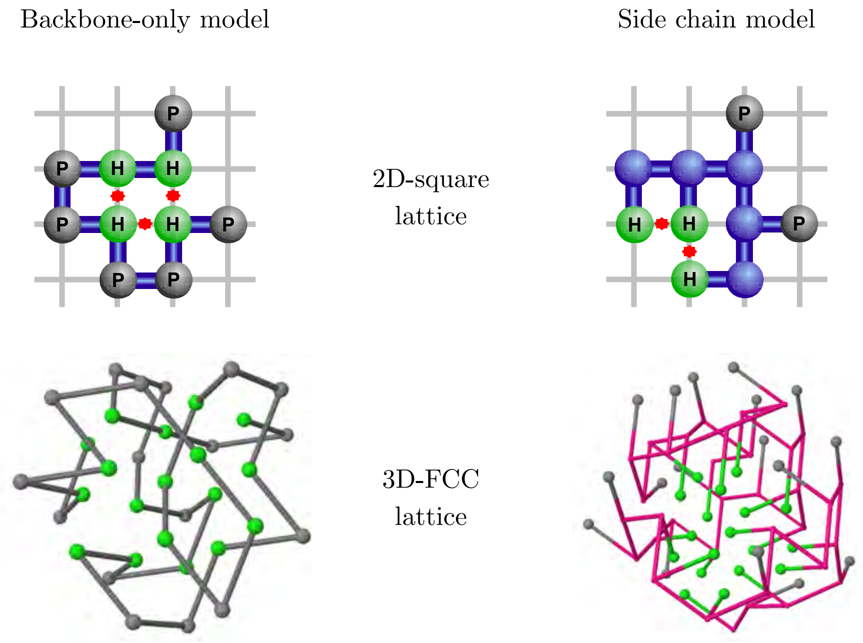


Figure 1.7: Comparison of backbone-only and side chain lattice protein models in different lattices.

increasing sequence length n is depicted. Note that even for very short sequence lengths more than a million non-symmetric structures are possible despite the simplicity of the model, e.g. for sequence length 9 in the 3D-FCC lattice. Nevertheless, even this fast growing structure space is still orders of magnitude smaller than less abstract protein models. Therefore, backbone-only lattice protein models are often used to perform large scale protein studies.

Backbone-only models lack a side chain representation. Therefore, no space is reserved for side chains such that the backbone monomers can be placed closer together than as it would be possible in real structures. This usually results in too compact structures when compared to real proteins. A possible solution is the application of sophisticated energy functions that try to compensate for the effect (*Miyazawa and Jernigan*, 1996) or the use of side chain models.

1.2.2.2 Side Chain Models

Within side chain models both the backbone and side chain of each amino acid are represented by a monomer within the lattice. The backbone monomer represents the C_α-/backbone-atom while the side chain monomer abstracts the side chain's centroid, i.e. its geometric center, or the side chain's center of mass (*Bromberg and Dill*, 1994). Again, both monomers are without volume or mass and have to be neighbored within the lattice. "This simple representation approximates the frequency weighted (*Karlin et al.*, 1992) ratio of side-chain to main-chain volumes of the amino acid residues found in proteins (*Finney*, 1975)."[3] Examples of side chain models in different lattices are given in Fig. 1.5 and on the right of Fig. 1.7.

Given a lattice L, a *side chain lattice protein structure* P of length n is a sequence of coordinate pairs $P = ((P^b_1, P^s_1), \ldots, (P^b_n, P^s_n))$ where P^b_i and P^s_i denote the backbone and side chain monomer position of the i-th amino acid, respectively. All occupied positions have to be different, while

[3] Taken from (*Bromberg and Dill*, 1994)

	backbone-only model				**side chain model**			
n	SQR	CUB	FCC	210	SQR	CUB	FCC	210
2	1	1	1	1	5	6	32	289
3	2	2	4	15	25	73	2,867	134,210
4	5	6	32	289	119	1,159	257,855	66,489,324
5	13	22	313	6,223	512	19,402	22,432,448	
6	36	92	3,196	139,777	2,149	328,085		
7	98	402	32,835	3,172,350	8,846	5,516,455		
8	272	1,832	337,056		36,241			
9	740	8,453	3,452,392		147,878			
10	2,034	39,640			601,677			
11	5,513	186,296			2,442,868			
12	15,037	881,147						
13	40,617	4,162,866						
14	110,188							
15	296,806							
16	802,075							
17	2,155,667							
$\sim$	$0.2764 \cdot$	$0.0765 \cdot$	$0.0494 \cdot$	$0.0340 \cdot$	$1.6106 \cdot$	$0.3026 \cdot$	$0.3692 \cdot$	$0.5836 \cdot$
$c \cdot b^n$	2.693^n	4.424^n	9.565^n	21.481^n	4.133^n	16.214^n	88.048^n	485.764^n

Table 1.2: The number of possible lattice protein structures for varying length n in the different lattices from Tab. 1.1. Note, the numbers exclude symmetric structures (resulting from rotation or reflection) and are given up to the length where a million structures is exceeded. In the bottom line an exponential formula of the form $c \cdot b^n$ is given to approximate the numbers for a given n in each lattice. It was obtained by linear regression on the available log data as explained in Fig. 1.8.

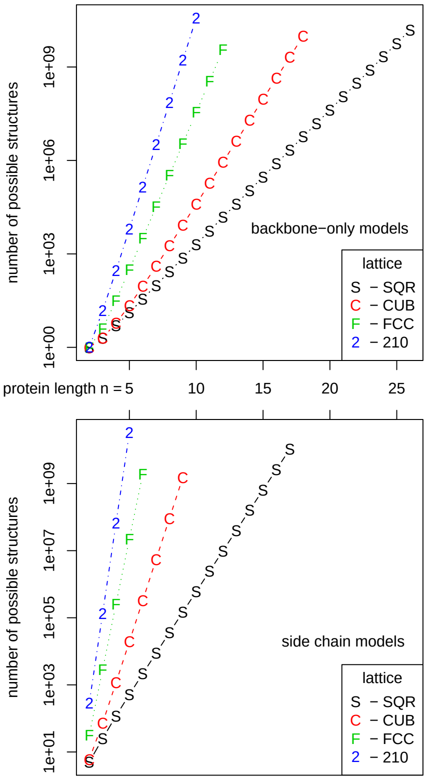


Figure 1.8: Exponential growth in the possible number of structures of sequence length n. Note, the numbers are given in logarithmic scale (Y-axis) and thus resulting in an almost linear function. Assuming an exponential growth of the form $y = c \cdot b^n$ we get the linear function by $\log_{10}(y) = \log_{10}(c) + n \cdot \log_{10}(b)$. Using linear regression we can get approximate values for $\log_{10}(c)$ and $\log_{10}(b)$. The resulting b, c values and $c \cdot b^n$ functions are shown in Tab. 1.2. The table depicts explicit structure numbers as well.

according backbone and side chain monomers as well as successive backbone monomers have to be neighbored in the lattice, i.e.

$$\forall_{1\leq i\leq n}: \quad P_i^b, P_i^s \in L \tag{1.11}$$

$$\forall_{1\leq i<j\leq n}: \quad P_i^b \neq P_j^b \wedge P_i^s \neq P_j^s \quad \text{i.e. } P^b \text{ and } P^s \text{ are self-avoiding} \tag{1.12}$$

$$\forall_{1\leq i,j\leq n}: \quad P_i^b \neq P_j^s \quad \text{i.e. } P^b \text{ vs. } P^s \text{ is self-avoiding} \tag{1.13}$$

$$\forall_{1\leq i\leq n}: \quad (P_i^s - P_i^b) \in N_L \quad \text{i.e. side chain } P_i^s \text{ is neighbored to backbone } P_i^b \tag{1.14}$$

$$\forall_{1<i\leq n}: \quad (P_i^b - P_{i-1}^b) \in N_L \quad \text{i.e. backbone } P_i^b \text{ is neighbored to predecessor } P_{i-1}^b \tag{1.15}$$

Side chain lattice proteins are more realistic due to the explicit representation of the amino acid's side chain. It prevents unnatural collapse during structural studies (*Eyrich et al.*, 1999) and enables the reconstruction of full atom protein data (*Feig et al.*, 2000). This gain in realism is paid for by increased complexity, since the structure space in side chain models growths much faster than for backbone-only models as shown in Tab. 1.2 and Fig. 1.8. For instance within the chess knights walk (210) lattice, the number of possible structures exceeds already 66 million for sequence length 4 compared to 289 structures for backbone-only models. This exemplifies the vast increase in computational complexity when incorporating side chains into protein models.

1.2.2.3 Structure Encoding and Symmetry Breaking

In order to enable a compact and handy lattice protein structure representation, so called *move strings* can be used. In general, a move string is a character string encoding of the neighboring vectors within the protein structure along the chain. To this end a unique mapping of neighboring vectors N_L to an arbitrary *move alphabet* Σ_M is needed for each lattice, i.e. $m : N_L \to \Sigma_M$. The *move string* M of a protein structure P of length n is thus defined by

$$\begin{aligned} \text{backbone-only models} \quad &: \quad M(P) = (M_1, \ldots, M_{n-1}) \in \Sigma_M^{n-1} \\ &\text{with} \quad M_i = m(P_{i+1} - P_i) \end{aligned} \tag{1.16}$$

$$\begin{aligned} \text{side chain models} \quad &: \quad M(P) = (M_1, \ldots, M_{2n-1}) \in \Sigma_M^{2n-1} \\ &\text{with} \quad M_i = \begin{cases} m(P^s_{(i+1)/2} - P^b_{(i+1)/2}) & \text{if } i \text{ odd (backbone} \to \text{side chain)} \\ m(P^b_{(i/2)+1} - P^b_{(i/2)}) & \text{if } i \text{ even (backbone} \to \text{backbone)} \end{cases} \end{aligned} \tag{1.17}$$

A common move alphabet Σ_M used within this manuscript is the encoding of the coordinate changes with according intuitive directions, i.e. changes in X-axis are encoded by Forward/Backward, in Y-axis by Left/Right, and in Z-axis by Up/Down. For example the neighboring vector $(0, 0, 1) \in N_{\text{CUB}}$ in the cubic lattice would be mapped to $m((0, 0, 1)) = \texttt{U}$, where in the FCC lattice $(1, -1, 0) \in N_{\text{FCC}}$ is mapped to $m((1, -1, 0)) = \texttt{FL}$. Note, since two coordinates are changed within the FCC lattice, the mapping results in a two letter alphabet. A straightforward algorithm to get a string representation of a given (neighboring) vector embeddable in the 3D-cubic lattice is shown in Alg. 1.

Algorithm 1 Algorithm to obtain a string encoding for a given vector in a lattice. It is based on a move string encoding $m : N_{\text{CUB}} \to \Sigma_M$ that maps the neighboring vectors of the 3D-cubic lattice onto some move alphabet Σ_M, e.g. $\Sigma_M = \{\texttt{F,B,L,R,U,D}\}$. Furthermore, string concatenation is denoted by $\oplus$.

```
function VectorToString($\vec{v} \in L_{\text{CUB}}$)
given $\vec{v} = (X, Y, Z)$
    return $m(\ (X, 0, 0)\ )^{|X|} \ \oplus\ m(\ (0, Y, 0)\ )^{|Y|} \ \oplus\ m(\ (0, 0, Z)\ )^{|Z|}$
end function                              ▷ i.e. adds an element from $\Sigma_M$ for each move per axis
```

To ease the readability of move strings in side chain models, we put the encoding of the side chain neighboring in parentheses. This results for instance in the cubic lattice in a move string like “`(U)R(F)B(L)`” since we have one neighboring vector to side chains more than backbone connections. Note, when ignoring the parentheses and their content the backbone-only move string of the structure can be obtained, i.e. “`RB`”.

Since we encode the absolute neighboring vectors along the chain, such move strings are also called *absolute move strings.* They enable a translation independent structure representation, since they are independent from the 3D-position of the first monomer that serves as the starting point of the encoding. Unfortunately, they are sensitive to symmtries via rotation or reflection. As an example consider the backbone-only structures `FFF` and `BBB`. They are encoding the same structure which is a straight chain within the lattice pointing in positive and negative direction along the X-axis, respectively. Thus, they are representing the same structure just reflected on the Y-axis or rotated by 180 within the X-Y- or X-Z-plane.

A possible workaround to enable a rotation-independent structure representation is the usage of *relative move strings* (*Bornberg-Bauer*, 1997c). Here, the relative movements along the chain are encoded. For instance an absolute move string `LFRR` would be described by `FRRF` following the direction changes along the structure. The relative move string encoding is based on explicit coordinate system rotations via the multiplication with according rotation matrices, see (*Backofen et al.*, 2000). The latter makes the calculation of relative move strings computationally expensive.

In the following, we will introduce a faster approach to get a rotation and reflection independent structure representation using absolute move strings. To this end, we will normalize our move strings via the unique selection of a representative move string among all symmetric ones.

A *symmetry* r is a total injective function $r : \Sigma_M \longrightarrow \Sigma_M$ that maps each neighboring vector onto its symmetric equivalent due to rotation or reflection. Thus, such a symmetry r corresponds to the product of a symmetry-specific rotation/reflection matrix R_r with the encoded neighboring vector $\vec{n} \in N_L$ such that $r(m(\vec{n})) = m(\vec{n}R_r)$. Since we have a fixed neighborhood N_L and move encoding Σ_M for a given lattice, we can do a tabularization of the symmetric mappings serving as a fast lookup of the symmetry functions r. With $\mathcal{R}$ we denote the set of all such symmetries operating on Σ_M, which is closed under inversion, i.e. $r \in \mathcal{R} \rightarrow r^{-1} \in \mathcal{R}$.

As an example consider the 2D square lattice and a structure reflection in the X-axis. Thus, the Y-axis neighboring vectors `L,R` change and are mapped onto each other with reverse symmetry, i.e. $r(\texttt{L}) = \texttt{R}$ and $r(\texttt{R}) = \texttt{L}$, whereas the X-axis neighboring vectors `F,B` are retained, i.e. $r(\texttt{F}) = \texttt{F}$ and $r(\texttt{B}) = \texttt{B}$. According symmetries are defined for all possible symmetric rotations or reflections.

The set of all symmetries $\mathcal{R}$ can now be used to normalize a move string M, i.e. to convert it into a unique move string for all symmetries. This is done by translating M into the lexicographically smallest move string possible for all symmetries (using symmetry $\check{r} \in \mathcal{R}$) such that

$$\forall_{r \in \mathcal{R}} \ : \ (\check{r}(M_1), \ldots, \check{r}(M_n)) \ \leq_{lex} \ (r(M_1), \ldots, r(M_n)) \ . \tag{1.18}$$

Since the number of symmetries is usually small, this enables fast symmetry detection. Within the 2D square lattice 8 symmetries are possible; the 3D cubic and FCC lattice can have 48 symmetries.

In Table 1.2 the numbers of non-symmetric structures are shown. They were calculated by full enumeration of all possible normalized move strings for valid structures. The number of all possible structures including symmetric ones is orders of magnitude higher. For instance there are 2.15 million non-symmetric structures of length 17 in the simple 2D square lattice (Tab. 1.2). When allowing for the 8 possible symmetries this number increases to 46.47 million including symmetric ones (*Sykes et al.*, 1972). In the 3D cubic lattice the 4.16 million structures of length 13 raise up to 943.97 million respectively (*Sykes et al.*, 1972).

Gan et al. (2008) have shown, that the exclusion of symmetric structures within folding simulations can significantly increase the search speed for low energy conformations. To this end, the introduced move string normalization can be applied as a fast filter to remove or ignore symmetric structures. Furthermore, *Backofen and Will* (2002) have introduced a framework to enable a symmetry breaking within constraint-based protein structure prediction (CPSP) in 3D HP-models (*Backofen and Will*, 2006). This Constraint Programming extension is applied within all CPSP applications within this thesis and enables the enumeration of non-symmetric optimal structures as later discussed in Chapter III.

1.2.2.4 Other Lattice-based Models

Extending the standard backbone-only model, *Ponty et al.* (2008) implemented a full atom backbone model (see Fig. 1.4). Here, all atoms of the protein backbone are represented by monomers and confined to neighbored lattice nodes within the 3D cubic or FCC lattice.

The CABS protein model introduced by Kolinski and co-workers (*Ekonomiuk et al.*, 2005), uses a lattice to discretize the C_α backbone trace of a protein. In contrast to other models, the CABS model uses a large set of 800 neighboring vectors of various lengths (based on cubic lattice) to define the possible relative placements of C_α monomers. To represent side chains up to two monomers are used, one representing the C_β atom and one the center of the side group. Note, in contrast to the side chain lattice model from above, both monomers are placed *off-lattice*, i.e. they are not confined to the lattice. Their relative positions are derived from the statistics of known protein structures and only the most probable ones are allowed. In addition, the center of the C_α-C_α peptide bond is represented off-lattice, too. Another difference to the previously introduced lattice protein models is the explicit volume representation of each monomer. Furthermore, it utilizes an advanced energy function to model the force fields of the physical interactions in real proteins. A schematic illustration of a short fragment of a CABS model is given in Fig. 1.4.

1.2.3 Abstraction of Sequence Space and Free Energy

Functional proteins are assumed to be in thermodynamic equilibrium (*Finkelstein and Badretdinov*, 1997), i.e. they form a structure with minimal free energy. Thus, if we are able to calculate the minimal energy structure of a given protein, we should be able to infer its function since function is mediated by structure (see Sec. 1.1).

In order to model thermodynamics of the folding process of proteins, an abstraction of a structure's free energy within the system is required. Due to the vast structure simplifications in lattice proteins, it is not possible (nor intended) to model all possible *in vivo* folding force fields listed in Sec. 1.1. In lattice protein models free energy calculations are often reduced to simple pairwise potentials (*Lau and Dill*, 1989; *Kolinski and Skolnick*, 2004; *Miyazawa and Jernigan*, 1985). Thus, given a protein of length n with amino acid sequence S and structure P, the energy E of the protein is calculated by

$$\text{backbone-only model} \quad : \quad E(S,P) = \sum_{1 \leq i+1 < j \leq n} e(S_i, P_i, S_j, P_j) \tag{1.19}$$

$$\text{side chain model} \quad : \quad E(S,P) = \sum_{1 \leq i < j \leq n} e(S_i, P_i^s, S_j, P_j^s) \tag{1.20}$$

where $e(S_i, P_i, S_j, P_j)$ denotes the pairwise potential for the i-th and j-th amino acid. Note, Eq. 1.19 for backbone-only models ignores energy contributions for consecutive monomers. Note further, in side chain models only side chain interactions (P^s) are evaluated in Eq. 1.20.

Based on the interaction range considered for pairwise energy contributions, one distinguishes *contact- and distance-based energy functions.*

A *contact-based function* considers a contribution only if the two amino acids are within a maximal distance $\delta_{\max}$. Thus, the pairwise potentials only depend on the possible amino acids in S, i.e.

it can be represented as a function

$$\begin{aligned} e_S \quad &: \quad \Sigma_S \times \Sigma_S \to \mathbb{R} & (1.21) \\ \text{such that} & & \\ e(S_i, P_i, S_j, P_j) \quad &= \quad \begin{cases} e_S(S_i, S_j) & \text{if } |P_i - P_j| \leq \delta_{\max} \\ 0 & \text{else} \end{cases} & (1.22) \end{aligned}$$

where Σ_S denotes the model's alphabet of amino acids, i.e. $S \in (\Sigma_S)^n$. Usually, the maximal distance $\delta_{\max}$ is set to the length of the neighboring vectors within the model, i.e. $\delta_{\max} = |\vec{v}|$ with $\vec{v} \in N_L$. Thus, only amino acid pairs neighbored within the lattice contribute to the contact-based energy function. Examples are the HP model by *Lau and Dill* (1989) or the potentials by *Miyazawa and Jernigan* (1996), both discussed later.

In order to incorporate more global and detailed folding forces *distance-based energy functions* were applied (see a review by *Hart and Newman* (2006)). Here, either sequence based potentials (as e_S) are scaled by the distance (*Unger and Moult*, 1993), or special distance dependent potentials are applied (*Ngo and Marks*, 1992).

In this manuscript only contact-based energy functions are considered and the applied potentials e_S are introduced next.

1.2.3.1 HP-model

The **H**ydrophobic-**P**olar (HP) model was first introduced by *Lau and Dill* (1989) to describe water-soluble globular proteins. Its energy function focuses on hydrophobic interactions that are known to have a large influence on protein folding and structure (*Dyson et al.*, 2006). Thus, it abstracts from all the possible attracting and repulsing forces by considering only two classes of amino acids: hydrophobic (H) and hydrophilic/polar (P). The resulting energy function is then given by

$$e_S^{\text{HP}}(S_i, S_j) = \begin{cases} -1 & \text{if } S_i \textit{ and } S_j \text{ are } \textit{both} \text{ hydrophobic (H)} \\ 0 & \text{else} \end{cases} \tag{1.23}$$

Equation 1.23 shows that only interactions between hydrophobic residues, so called *HH-contacts*, are relevant for the energy calculation. All other interaction types (HP- or PP-contacts) result in no energy contribution. Structures with low energy show therefore a close packing of hydrophobic residues, usually resulting in a globular structure where hydrophobic monomers are gathered in its center. This phenomenon is intended and does reproduce the *hydrophobic cores* observed in real globular protein structures (*Munson et al.*, 1996).

Since the energy evaluation focuses on hydrophobicity only, the used sequence alphabet can be reduced to an assignment if an amino acid is hydrophobic ($\mathtt{H}$) or polar ($\mathtt{P}$), i.e. $\Sigma_S = \{\mathtt{H}, \mathtt{P}\}$. Throughout this thesis the classification follows *Ullah et al.* (2009) and is given in Tab. 1.3. Other hydrophobicity classification and assignment schemes are possible (e.g. by *Sandelin* (2004)) but not considered within this manuscript.

The HP model is one of the simplest protein models possible. Nevertheless sequences/structures have been shown to be protein-like with respect to several properties: Sandelin and co-workers

Hydrophobic (**H**)			Hydrophilic/Polar (**P**)					
C	:	Cysteine	A	:	Alanine	Q	:	Glutamine
F	:	Phenylalanine	D	:	Aspartic Acid	R	:	Arginine
I	:	Isoleucine	E	:	Glutamic Acid	S	:	Serine
L	:	Leucine	G	:	Glycine	T	:	Threonine
M	:	Methionine	H	:	Histidine			
V	:	Valine	K	:	Lysine			
W	:	Trytophan	N	:	Asparagine			
Y	:	Tyrosine	P	:	Proline			

Table 1.3: The used **H**ydrophobic-**P**olar classification of amino acids take from (*Ullah et al.*, 2009). It is used to calculate the energy function e_S^{HP} from Eq. 1.23 and to convert an amino acid sequence into an HP sequence. Each amino acid is given with its single letter code.

found similar fluctuations of hydrophobicity within HP-model and real protein sequences (*Irbäck and Sandelin*, 2000; *Sandelin*, 2004) where studies by *Dill et al.* (1995) and *Guo et al.* (2007) discuss comparable surface to core ratios. Furthermore, HP-structures show protein-like secondary structures and repeating motifs (*Helling et al.*, 2001; *Wroe et al.*, 2005) as well as alike volume exclusion among residues (*Xia and Levitt*, 2004b). In addition, the properties of their hydrophobic cores and polar exteriors are in agreement with real protein structures (*Sandelin*, 2004). The model itself is particularly useful in studies that require extended coverage of both sequence and structure space (*Wroe et al.*, 2005).

Despite its simplicity, the computational problems within the model remain hard. *Berger and Leighton* (1998) have shown the NP-completeness of the protein structure prediction problem while *Berman et al.* (2007) have proven the NP-completeness of the inverse folding problem in HP models. Both problems are discussed in Sec. 1.3.

1.2.3.2 MJ-model

Miyazawa and Jernigan (1985) introduced amino acid specific pairwise potentials to be applied within protein folding simulation. The resulting energy function became a standard for lattice protein studies using the full 20 amino acid sequence alphabet. The potentials are derived from real protein crystal structures and result in attractive inter-residue energies as well as repulsive energy terms to avoid a too dense packing (*Miyazawa and Jernigan*, 1996). For each amino acid type the solvent accessibility is estimated and incorporated into the potentials.

The according pairwise energy contributions e_S^{MJ} are defined by the symmetric potentials MJ listed in Tab. 1.4, i.e.

$$e_S^{\mathrm{MJ}}(S_i, S_j) = \mathrm{MJ}_{S_i,S_j}. \tag{1.24}$$

The MJ energy model allows for a more detailed energy evaluation of protein structures than the much simpler HP model. But due to the distance independent energy contributions, its application is usually restricted to lattice protein models.

	C	M	F	I	L	V	W	Y	A	G	T	S	Q	N	E	D	H	R	K	P
C	-1.06	0.19	-0.23	0.16	-0.08	0.06	0.08	0.04	0.00	-0.08	0.19	-0.02	0.05	0.13	0.69	0.03	-0.19	0.24	0.71	0.00
M	0.19	0.04	-0.42	-0.28	-0.20	-0.14	-0.67	-0.13	0.25	0.19	0.19	0.14	0.46	0.08	0.44	0.65	0.99	0.31	0.00	-0.34
F	-0.23	-0.42	-0.44	-0.19	-0.30	-0.22	-0.16	0.00	0.03	0.38	0.31	0.29	0.49	0.18	0.27	0.39	-0.16	0.41	0.44	0.20
I	0.16	-0.28	-0.19	-0.22	-0.41	-0.25	0.02	0.11	-0.22	0.25	0.14	0.21	0.36	0.53	0.35	0.59	0.49	0.42	0.36	0.25
L	-0.08	-0.20	-0.30	-0.41	-0.27	-0.29	-0.09	0.24	-0.01	0.23	0.20	0.25	0.26	0.30	0.43	0.67	0.16	0.35	0.19	0.42
V	0.06	-0.14	-0.22	-0.25	-0.29	-0.29	-0.07	0.02	-0.10	0.16	0.25	0.18	0.24	0.50	0.34	0.58	0.19	0.30	0.44	0.09
W	0.08	-0.67	-0.16	0.02	-0.09	-0.07	-0.12	-0.04	-0.09	0.18	0.22	0.34	0.08	0.06	0.29	0.24	-0.12	-0.16	0.22	-0.28
Y	0.04	-0.13	0.00	0.11	0.24	0.02	-0.04	-0.06	0.09	0.14	0.13	0.09	-0.20	-0.20	-0.10	0.00	-0.34	-0.25	-0.21	-0.33
A	0.00	0.25	0.03	-0.22	-0.01	-0.10	-0.09	0.09	-0.13	-0.07	-0.09	-0.06	0.08	0.28	0.26	0.12	0.34	0.43	0.14	0.10
G	-0.08	0.19	0.38	0.25	0.23	0.16	0.18	0.14	-0.07	-0.38	-0.26	-0.16	-0.06	-0.14	0.25	-0.22	0.20	-0.04	0.11	-0.11
T	0.19	0.19	0.31	0.14	0.20	0.25	0.22	0.13	-0.09	-0.26	0.03	-0.08	-0.14	-0.11	0.00	-0.29	-0.19	-0.35	-0.09	-0.07
S	-0.02	0.14	0.29	0.21	0.25	0.18	0.34	0.09	-0.06	-0.16	-0.08	-0.20	-0.14	-0.14	-0.26	-0.31	-0.05	0.17	-0.13	0.01
Q	0.05	0.46	0.49	0.36	0.26	0.24	0.08	-0.20	0.08	-0.06	-0.14	-0.14	0.29	-0.25	-0.17	-0.17	-0.02	-0.52	-0.38	-0.42
N	0.13	0.08	0.18	0.53	0.30	0.50	0.06	-0.20	0.28	-0.14	-0.11	-0.14	-0.25	-0.53	-0.32	-0.30	-0.24	-0.14	-0.33	-0.18
E	0.69	0.44	0.27	0.35	0.43	0.34	0.29	-0.10	0.26	0.25	0.00	-0.26	-0.17	-0.32	-0.03	-0.15	-0.45	-0.74	-0.97	-0.10
D	0.03	0.65	0.39	0.59	0.67	0.58	0.24	0.00	0.12	-0.22	-0.29	-0.31	-0.17	-0.30	-0.15	0.04	-0.39	-0.72	-0.76	0.04
H	-0.19	0.99	-0.16	0.49	0.16	0.19	-0.12	-0.34	0.34	0.20	-0.19	-0.05	-0.02	-0.24	-0.45	-0.39	-0.29	-0.12	0.22	-0.21
R	0.24	0.31	0.41	0.42	0.35	0.30	-0.16	-0.25	0.43	-0.04	-0.35	0.17	-0.52	-0.14	-0.74	-0.72	-0.12	0.11	0.75	-0.38
K	0.71	0.00	0.44	0.36	0.19	0.44	0.22	-0.21	0.14	0.11	-0.09	-0.13	-0.38	-0.33	-0.97	-0.76	0.22	0.75	0.25	0.11
P	0.00	-0.34	0.20	0.25	0.42	0.09	-0.28	-0.33	0.10	-0.11	-0.07	0.01	-0.42	-0.18	-0.10	0.04	-0.21	-0.38	0.11	0.26

Table 1.4: The MJ-potentials by *Miyazawa and Jernigan* (1985) that are used within this manuscript. They are available within the LatPack package (*Mann et al.*, 2008a).

1.2.3.3 Other Models

In the last decade a wide variety of energy models has been suggested and a few are discussed in the following.

Concerning the energy contributions in the HP model different pairwise potentials have been used. Beneath the presented potentials by *Dill et al.* (1995) other values have been applied by *Bornberg-Bauer* (1997a) or *Li et al.* (1996).

Banavar et al. (2004) suggested the **THP** model to incorporate context specific HH-contact contributions. Here contacts in parallel structures are more weighted than "edge"-contacts.

The **HPNX** model, introduced in (*Bornberg-Bauer*, 1997a; *Backofen et al.*, 1999), distinguishes four different amino acid groups, namely **H**ydrophobic, **P**ositively charged, **N**egatively charged, and **X** for all remaining neutral residues. Still, HH-contacts are the strongest potentials in the energy model while the suggested energy contributions are different in (*Bornberg-Bauer*, 1997a) and (*Backofen et al.*, 1999).

An improved version, the **hHPNX** model by *Hoque et al.* (2009), follows the amino acid grouping of the **YhHX** model suggested by *Crippen* (1991) where Alanine and Valine are treated as special group (**h**). *Bornberg-Bauer* (1997a) introduced an integer conversion of the real valued **YhHX** model that maintains approximately the same ratios of entries. Unfortunately, some reported values were not correct and fixed by *Hoque et al.* (2009).

Work for pairwise potentials between all 20 amino acids was done by *Berrera et al.* (2003). They introduced a more simplistic potential extraction from real protein data compared to the MJ model by *Miyazawa and Jernigan* (1985).

1.3 Applications and Problems

Before going into details, the problems studied within this thesis along with some applications of protein models are briefly introduced. Formal definitions and a more detailed introductions are given in the according referenced chapters.

1.3.1 Protein Chain Lattice Fitting

In order to evaluate the applicability of different lattices for protein models, a representative lattice protein structure has to be calculated. For a given full atom protein structure the best structure representation within the lattice model must be found, i.e. the structure that minimizes the applied distance measure. A graphical sketch is given in Fig. 1.9

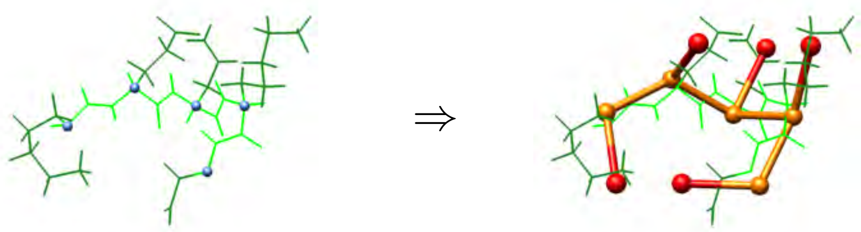


Figure 1.9: The Protein Chain Lattice Fitting (PCLF) problem is to calculate the optimal lattice protein model (backbone/sidechain in orange/red) of a protein (backbone/sidechain in light-/darkgreen; C_α-positions in blue) given in full atom representation minimizing some distance function.

Maňuch and Gaur (2008) have shown the NP-completeness of this problem for backbone-only 3D-cubic lattice models when minimizing coordinate root mean square deviation (cRMSD) and named it the *Protein Chain Lattice Fitting (PCLF) problem*. Here exemplified for dRMSD-minimization that is introduced in Sec. 2.2.

In Chapter II we will introduce a new chain-growth-based approach to the PCLF problem that enables a fast and approximate lattice model fitting (Sec. 2.2). Furthermore, we introduce a constraint-based refinement protocol that enables the optimization of such approximated solutions within Sec. 2.3.

1.3.2 Optimal Structure Prediction

The *Protein Structure Prediction (PSP) problem* is to determine the optimal protein structure for a given protein sequence within the model. A structure is optimal if it minimizes the energy function of the model. The problem is sketched for the HP model in Fig. 1.10.

The PSP problem is a hard computational problem. *Unger and Moult* (1993) have shown that the problem is NP-hard for arbitrary pairwise energy functions. Even in the simple HP models it was shown to be NP-complete, both in the 2D-square (*Crescenzi et al.*, 1998) and the 3D-cubic lattice (*Berger and Leighton*, 1998).

Backofen and Will (2006) have introduced the Contraint-based Protein Structure Prediction (CPSP) approach to solve the PSP in backbone-only 3D HP models which is discussed and extended in Chapter III. Therein we briefly introduce the CPSP approach and present its extension

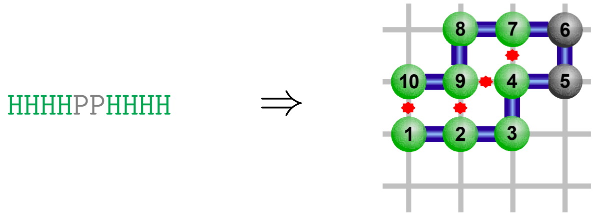


Figure 1.10: The Protein Structure Prediction (PSP) problem is to calculate the optimal (energy minimizing) protein structure within the protein model (right) given only the protein's sequence (left). Here it is shown for the backbone-only HP model in the 2D-square lattice while formed HH-contacts are highlighted in red. H-monomers are given in green, P-monomers in gray, and the backbone connections in blue.

to side-chain models (Sec. 3.2). Using the resulting tools from our CPSP-tools package (Sec. 3.2.3), we undertake an extensive degeneracy study in both backbone-only and side chain models in Sec. 3.3. The immense degeneracy of optimal structures in the HP model triggers the need for a grouping of optimal structures. In Section 3.4 we present to this end a hydrophobicity-focusing equivalence relation for optimal structures and introduce a CPSP-based approach to identify the different equivalence groups. Their number is shown to be orders of magnitude smaller than a sequence's degeneracy. The equivalence definition can be used in evolutionary studies as discussed in Sec. 6.4. Furthermore, the much lower number of such equivalence classes is of high importance when we want to use optimal structures from the HP model as compact start conformations for local search schemes for full 20 amino acid energy functions. Such a hybrid approach is introduced in Sec. 3.5 and is shown to outperform classic local search protocols.

1.3.3 Folding Simulation

The goal of *Folding Simulations* is to mimic the structure formation process of a protein. Thus, one is interested in the structural changes over time rather than only the final result (as in the PSP).

For discretized protein models, like lattice models, often Markov Chains (introduced in the next section) are used to simulate the folding. In order to enable a reasonable representation of the underlying chemical and physical processes, only small structural differences between successive structures are usually allowed. Thus, the definition of the allowed changes via so called move sets is central to the efficiency and realism of the folding simulation. Figure 1.11 exemplifies a few steps within a folding simulation.

In Chapter IV a generic k-local move set definition for side chain models is introduced. It is, to our knowledge, the first such generic definition in literature that is applicable within any lattice for arbitrary interval lengths k. We present a constraint-based implementation of the move set in Sec. 4.3 that preserves the generality of the move set and exemplify its application in Sec. 4.4.

Independently, we perform an exhaustive study of co-translational folding, i.e. the folding process of the premature protein while it is assembled at the ribosome, in Chapter V. To this end, we introduce a co-translational folding simulation scheme (Sec. 5.2) and a low-energy-pathway-based classification protocol (Sec. 5.3). These are used in Sec. 5.4 to generate large data sets of sequences that can reach their native structure co-translationally and negative sets of sequences that can not. This enables the study of sequence and structure properties that are connected to co-translational

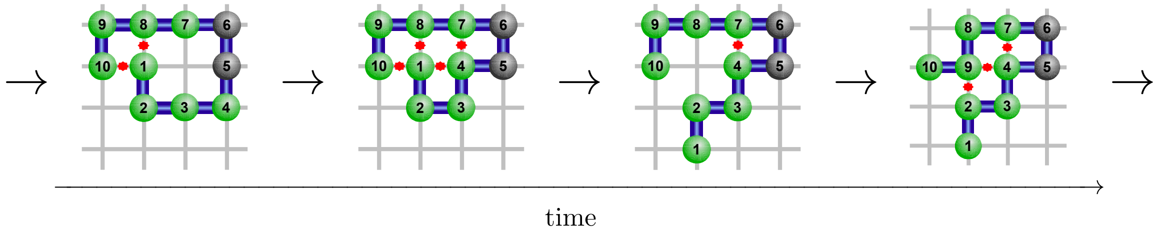


Figure 1.11: Folding Simulations are time series of protein structures that aim to mimic the native structure formation process. The figure exemplifies a time series for the backbone-only HP model in the 2D-square lattice (light gray) where energy contributing HH-contacts are highlighted in red. H-monomers are given in green, P-monomers in dark gray, and the backbone connections in blue.

folding. Where possible, we lift our result to real proteins.

Furthermore, we introduce a new protocol to define and identify protein-like sequences in protein models in Sec. 5.5. It is exemplified in the 3D HP model and incorporates folding simulations, optimal structure prediction, and co-translational folding.

1.3.4 Inverse Folding / Sequence Design

The *Inverse Folding* or *Sequence Design* problem is to determine a protein sequence for a given protein structure such that the combination minimizes a given optimization criterion. Often, a sequence is wanted that forms the given structure as its unique energy minimal one (*Hart*, 1997). Further criteria that constrain the number of H-monomers or the sequence's degeneracy are possible. See Fig. 1.12 for an illustration.

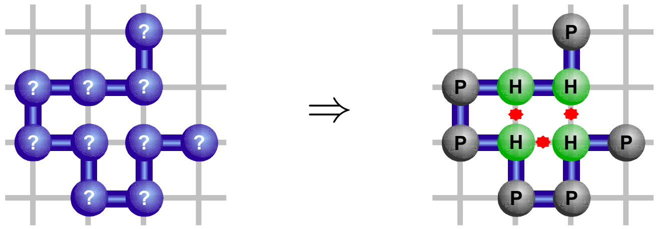


Figure 1.12: The Inverse Folding problem is to determine for a given structure (in blue on the left) a protein sequence, such that some optimization criteria is met. The problem is depicted for the backbone-only HP model in 2D-square lattice and the sequence has to adapt the given structure as an energy minimal one. This holds for the sequence presented on the right. H-monomers are given in green, P-monomers in gray, and HH-contacts are highlighted in red.

Hart (1997) has shown that the Inverse Folding problem is NP-hard in the 2D-square HP lattice models when requiring that the given structure is formed with minimal energy. The results were later revisited by *Berman et al.* (2007) and extended to the 3D-cubic lattice.

In Section 6.2 from Chapter VI a new approach to solve the inverse folding problem in 3D HP models is presented. The method is based on the screening of the structure with optimal H-monomer packings to identify candidate sequences that will form the given structure with high probability as their energy minimal one. The latter is evaluated using the CPSP approach from

Sec. 3.2, which also enables the investigation whether the structure is the unique ground state of the sequence or not.

This approach enables the study of neutral evolution in 3D HP models done in Chapter VI since we can design sequences that form a given structure as their unique ground state. These sequences form a neutral network in sequence space that models the possible pathways of neutral evolution. In Section 6.3 we perfom an extensive study of the abundance and features of such neutral networks in 3D models and compare to known results from 2D studies.

1.4 General Techniques

This thesis studies the problems and applications introduced in Sec. 1.3. The general techniques and approaches used are introduced within this section to enable a clear presentation and to reduce redundancy.

In Section 1.4.1 and 1.4.2, time discrete Markov Chains and Markov Chain Monte Carlo methods are introduced. These are essential for folding simulations and local search methods within the discrete structure space of lattice proteins. They are applied to find energy minimal structures (Sec. 3.5) or to simulate the folding process (Sec. 4.4, 5.2, and 5.5). Furthermore, they can be applied to traverse the sequence space when searching for sequences with low-degenerated ground states (Sec. 6.3).

Constraint Programming, introduced in Sec. 1.4.3, is a flexible framework to tackle hard computational problems. Therefore, it is used to predict optimal structures in the HP model as done in the CPSP approach and its extensions discussed in Chapter III. Furthermore, it is applied to refine lattice protein fits of real protein structures (Sec. 2.3) or to compute the local neighborhood of structures within structure space presented in Sec. 4.3.

1.4.1 Markov Chains

Markov Chains are random processes with no or very restricted history producing a series of discrete states. Within this thesis only 1^{st}*-order time-homogeneous Markov Chains* with no history are used, i.e. the conditional probability distribution of future states of the process depends only upon the present state. The latter is known as the *Markov Property.* Furthermore, we focus on *time-discrete* Markov Chains where we consider only discrete time points and enforce the random process to be in a distinct state at each time point. Throughout, we follow the notations of *Clote and Backofen* (2000).

Formally, such a Markov Chain is defined by the triple $\langle \mathbb{Q}, \pi, \mathbb{P} \rangle$. $\mathbb{Q}$ defines the finite set of states that can be adopted. To ease the presentation, states are represented by indices, i.e. $\mathbb{Q} = \{1, 2, \ldots\}$. The vector $\pi = (\pi_1, \ldots, \pi_{|\mathbb{Q}|})$ provides a start probability π_i for each state $i \in \mathbb{Q}$. Finally, the stochastic transition matrix $\mathbb{P}$ defines the transition probabilities $\mathbb{P}_{i,j}$ for each state $i \in \mathbb{Q}$ into any other state $j \in \mathbb{Q}$.

Given a time series of observed states $(o_0, \ldots, o_t) \in \mathbb{Q}^{t+1}$ produced by a Markov Chain it follows

$$\begin{aligned} \Pr(o_0 = i) &= \pi_i & (1.25) \\ \Pr(o_{t+1} = j \in \mathbb{Q} \mid o_0, \ldots, o_t = i \in \mathbb{Q}) &= \Pr(o_{t+1} = j \mid o_t = i) \\ &= \mathbb{P}_{i,j} & (1.26) \end{aligned}$$

where Eq. 1.26 defines the Markov Property. The element $\mathbb{P}^t_{i,j}$ of the t-th matrix exponential of $\mathbb{P}$ defines the transition probability between the two states in t steps, i.e.

$$\mathbb{P}^t_{i,j} = \Pr(o_t = j \mid o_0 = i) \ . \tag{1.27}$$

Based on that, the *period* of a state $i \in \mathbb{Q}$ is defined by the greatest common divisor of $\{t \mid \mathbb{P}^t_{i,i} > 0\}$. If the period of all states in $\mathbb{Q}$ is one, the Markov Chain is *aperiodic*.

A Markov Chain is *irreducible*, if it can connect any two states of $\mathbb{Q}$ with a non-zero transition probability (Eq. 1.28), i.e. one can transform any state into any other.

$$\forall_{i,j \in \mathbb{Q}} : \exists_{t \geq 1} : \mathbb{P}^t_{i,j} > 0 \ . \tag{1.28}$$

An aperiodic, irreducible Markov Chain exhibits a *stationary or equilibrium distribution* π^* given by

$$\pi^* = \lim_{t \to \infty} \pi \cdot \mathbb{P}^t \ . \tag{1.29}$$

A Markov Chain is said to be *reversible*, if there exists a stationary distribution π^* such that for any two states $i, j \in \mathbb{Q}$ the following Eq. 1.30 is fulfilled; a condition also known as *detailed balance*.

$$\pi^*_i \cdot \mathbb{P}_{i,j} \ = \ \pi^*_j \cdot \mathbb{P}_{j,i} \tag{1.30}$$

The conversion of Markov Chains into a distribution where detailed balance is fulfilled, the so called *steady state*, can be used to sample from that distribution. This is done by Markov Chain Monte Carlo methods, introduced next.

1.4.2 Markov Chain Monte Carlo Methods

To derive samples from a given probability distribution *Markov Chain Monte Carlo* methods can be used. They construct a reversible Markov chain that has the desired distribution as its equilibrium distribution. As discussed above, any start probability distribution for a reversible Markov Chain will converge to its stationary distribution (see Eq. 1.29). Thus, the state of the Markov Chain reached after a large number of steps can be assumed to be seen according to the equilibrium distribution. Since the latter equals the desired distribution, the reached state is then used as a sample from the distribution. The quality of the sample improves as a function of the number of steps (*Clote and Backofen*, 2000).

In the following some Markov Chain Monte Carlo algorithms that are used within the thesis are introduced.

1.4.2.1 Metropolis algorithm

Metropolis et al. (1953) introduced a method to obtain a sequence of random samples from a Boltzmann distribution. Subsequently, the algorithm was generalized by *Hastings* (1970).

Given an energy function $E : \mathbb{Q} \to \mathbb{R}$ that assigns an energy to each state of a Markov Chain $\langle \mathbb{Q}, \pi, \mathbb{P} \rangle$, the stationary distribution π^* is given by the *Boltzmann distribution*:

$$\begin{aligned} \pi_i^* &= \frac{e^{-E(i)/kT}}{Z} && (1.31) \\ Z &= \sum_{i \in \mathbb{Q}} e^{-E(i)/kT} && (1.32) \end{aligned}$$

where Z is the *canonical partition function* at constant *temperature* T using the Boltzmann constant k.

The Metropolis algorithm uses a *proposal density* to select possible successors of a given state $i \in \mathbb{Q}$. For finite Markov Chains, this can be replaced by a *neighboring system* $\mathcal{N}$ that satisfies the following properties (*Clote and Backofen*, 2000), where $j \in \mathcal{N}(i)$ means that j is in the neighborhood of i:

$$\begin{aligned} \forall_{i,j \in \mathbb{Q}} &: \; i \notin \mathcal{N}(i) \\ & \;\;\; j \in \mathcal{N}(i) \leftrightarrow i \in \mathcal{N}(j) \\ & \;\;\; |\mathcal{N}(i)| = |\mathcal{N}(j)| \\ \forall_{i \neq j \in \mathbb{Q}} &: \; \exists_{(q_1 = i, \ldots, q_m = j) \in \mathbb{Q}^m} : \forall_{1 < t \leq m} : q_t \in \mathcal{N}(q_{t-1}) \end{aligned}$$

The last property makes the neighboring system *ergodic* and ensures that the underlying Markov Chain is still irreducible, i.e. every state can be reached from any other state (see Eq. 1.28).

The acceptance probability for the transition from state $i \in \mathbb{Q}$ to a neighbor $j \in \mathcal{N}(i)$ randomly drawn from $\mathcal{N}(i)$ with probability $\frac{1}{|\mathcal{N}(i)|}$ is than defined at temperature T by the *Metropolis criterion* $b(T)$ given as follows

$$\begin{aligned} b_{i,j}(T) &= \begin{cases} 1 & \text{if } E(j) \leq E(i) \\ \frac{\pi_j^*}{\pi_i^*} = e^{-(E(j)-E(i))/kT} & \text{otherwise, i.e. } E(j) > E(i) \end{cases} \\ &= \min\left\{ 1 \,,\; e^{-(E(j)-E(i))/kT} \right\} \;. && (1.33) \end{aligned}$$

Thus, a transition to a state with lower energy is always accepted, while the transition to a state with higher energy is only accepted according to the probability defined by the temperature dependent Boltzmann weight of the energy difference. If the transition is rejected, the current state is maintained. Given the above, the Metropolis algorithm can be sketched in a very general scheme presented in Alg. 2.

Since lower energy states are always accepted, the algorithm easily converges towards a *local minimum* $\check{m} \in \mathbb{Q}$ of the energy function according to the neighboring system, where a local minimum

Algorithm 2 Metropolis algorithm for a Markov Chain $\langle \mathbb{Q}, \pi, \mathbb{P} \rangle$ with the Boltzmann distribution as its stationary distribution π^* at temperature T and an ergodic neighboring system $\mathcal{N}$.

```
$o_0 \leftarrow$ random state from $\mathbb{Q}$                              ▷ set initial state at random
for all ( $t \in [1, t_{\max}]$ ) do                            ▷ do until maximal number of steps is exceeded
    $o' \leftarrow$ random state from $\mathcal{N}(o_{t-1})$           ▷ draw at random a neighbored successor proposal
    $r \leftarrow$ random number in $[0, 1]$                           ▷ random number for acceptance check
    if ( $r \leq b_{o_{t-1},o'}(T)$ ) then                       ▷ apply Metropolis criterion (Eq. 1.33)
        $o_t \leftarrow o'$                                          ▷ accept the proposed neighbor
    else
        $o_t \leftarrow o_{t-1}$                                     ▷ reject the proposal and keep current state
    end if
end for
```

is a state $\check{m}$ such that $\forall_{j \in \mathcal{N}(\check{m})} : E(j) \geq E(\check{m})$. The possibility to accept energetically worse neighbors enables that such minima can be left, depending on the chosen temperature T. If T is high, the Metropolis criterion of Eq. 1.33 will tend to one, for low T values the acceptance probability will tend towards zero. This temperature dependence is used in Simulated Annealing, discussed next.

1.4.2.2 Simulated Annealing

Simulated Annealing was independently introduced by *Kirkpatrick et al.* (1983) and *Černý* (1985) as an optimization scheme to find minima of the energy function. In principle, Simulated Annealing is a Metropolis algorithm (Sec. 1.4.2.1) framed by a cooling scheme for the temperature parameter T.

As discussed above, the temperature used within the Metropolis criterion guides the transition probability to states with higher energy. Higher temperature results in a higher acceptance probability. Within Simulated Annealing, the temperature is decreased periodically such that the system gets "frozen" within a local minimum. *Geman and Geman* (1984) have shown that for exponentially slow temperature cooling schedules, Simulated Annealing correctly computes the *global minimum* $\check{m} \in \mathbb{Q}$ of the energy function with $\forall_{j \in \mathbb{Q}} : E(j) \geq E(\check{m})$.

1.4.2.3 Adaptive, Gradient, and Random Walk

The following algorithms also traverse the state space $\mathbb{Q}$ and can be seen as special variants of the Metropolis algorithm from Sec. 1.4.2.1 for extreme temperatures. Thus, in the following only (temperature independent) replacements of the metropolis criterion $b_{i,j}$ have to be defined.

An ***Adaptive Walk*** equals a Metropolis algorithm at infinitely low temperature. Therefore, only states with lower energy are accepted since the Metropolis criterion from Eq. 1.33 results in values tending to zero (see b^{ADP} in Eq. 1.34). Thus, an adaptive walk *always concludes at a local minimum.* Nevertheless, the length and final conformation of the walk depends on the chosen neighbors along the simulation (see Alg. 2). Adaptive walks can be used to sample local minima accessible in an energy decreasing way from a given start state.

$$b_{i,j}^{\text{ADP}} = \begin{cases} 1 & \text{if } E(j) \leq E(i) \\ 0 & \text{otherwise } (E(j) > E(i)) \end{cases} \tag{1.34}$$

A variant of the Adaptive Walk is the ***Gradient Walk***, where within each step the *energetically lowest neighboring state* is chosen (see Eq. 1.35). Thus, the algorithm follows a gradient abstraction within the discretized optimization space. In contrast to the former, the Gradient Walk length is deterministic[4], both in length and the final minimum reached. This results in a unique assignment of any state of the structure space to a specific local minimum.

$$b^{\text{GRD}}_{i,j} = \begin{cases} 1 & \text{if } E(j) \leq E(i) \text{ and } \forall_{j' \in \mathcal{N}(i)} : E(j) \leq E(j') \\ 0 & \text{otherwise (not gradient neighbor)} \end{cases} \tag{1.35}$$

Finally, a ***Random Walk*** can be seen as a Metropolis algorithm at positive infinite temperature. By that, the Metropolis criterion tends to one, even for energetically worse neighbors, such that all neighbors are equally accessible by the algorithm, i.e. $b^{\text{RND}}_{i,j} = 1$. Therefore, Random Walks can be used to sample random states from $\mathbb{Q}$.

1.4.3 Constraint Programming

Constraint Programming (CP) is a technique to describe the solution space of a problem in terms of constraints instead of formulating a dedicated algorithm that calculates them. Therein, a constraint describes a condition that is fulfilled by all solutions, e.g. "$x < y$" or "*backbone and side chain monomers are placed neighbored in the lattice*". So if an instance of the problem under consideration satifies all constraints it is identified as a solution.

The CP framework covers techniques and methods to model the solution space of the problem via constraints as well as to calculate solutions of the modeled problem. The use of constraints allows to drawn a conclusion from partial solutions that are already found and enables a fast and efficient solving. The main application area for CP are NP-complete combinatorial problems.

In the following we introduce how to model a problem's solution in terms of CP. The mechanisms to solve the problem and how to calculate/enumerate solutions are only sketched and not part of this thesis. For further reading on this topic refer to standard literature, e.g. (*Tsang*, 1993).

In CP, problem solutions are modeled in terms of *Constraint Satisfaction Problems (CSP)*. A *finite domain CSP* is defined by a triple $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, where

- $\mathcal{X} = \{X_1, \ldots, X_n\}$ is the finite set of n variables describing the problem,
- $\mathcal{D} = \{D_1, \ldots, D_n\}$ is a set of finite domains, where D_i describes the domain of X_i, and
- $\mathcal{C}$ is a set of constraints defining the relations between the variables in $\mathcal{X}$.

A constraint $C \in \mathcal{C}$ is imposed on a tuple of variables expressed by $X(C) = (X_1, \ldots, X_k)$ where the constraint over a k-tuple is called k-ary. The expressed relation between the variables can be given semantically as a subset $T(C)$ of the Cartesian product of the according domains, i.e. $T(C) \subseteq D_1 \times \ldots \times D_k$.

An *assignment* is a function $A : \mathcal{X} \to \bigcup \mathcal{D}$ such that $A(X_i) \in D_i$. A constraint C with $X(C) = (X_1, \ldots, X_k)$ *is satisfied by an assignment* A if the restriction of A to the constrained variables $X(C)$ is among its allowed tuples, i.e. $(A(X_1), \ldots, A(X_k)) \in T(C)$. A *solution* A_s of a CSP is an assignment that satisfies all constraints in $\mathcal{C}$.

In order to identify solutions A_s of a CSP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, an iterative scheme of search and reasoning is applied. During the search, the domain D_i of a selected variable $X_i \in \mathcal{X}$ is divided into two

[4] This holds only for non-degenerated energy functions and will be discussed later.

independent subsets $\grave{D}_i$ and $\acute{D}_i$ (with $\grave{D}_i \cup \acute{D}_i = D_i$ and $\grave{D}_i \cap \acute{D}_i = \emptyset$) resulting in two independent subproblems of the initial problem. Afterwards, each subproblem is treated separately by a propagation step. Therein, the split of the domain is propagated onto the other variables/domains, removing values from $\mathcal{D}$ that are in conflict with constraints in $\mathcal{C}$. The latter is iterated until either

a) a domain holds no values any more, i.e. no assignment/solution can be found, or

b) no further propagation can be done but no assignment was reached, or

c) an assignment was found that satisfies all constraints, i.e. a solution A_s was detected.

In case (a) the system detected a failure of the search in this branch and a backtracking is done to investigate the remaining subproblem(s) of the CSP. In the second case (b), at least one domain holds more than one entry. Thus, another search iteration is performed, resulting in a subsequent set of two independent subproblems to be checked by a reasoning step. In the last case (c) a solution was found. If interested in the enumeration of more/all solutions, a backtracking is done to search within the remaining subproblem(s) of the CSP for further solutions.

The efficiency of this solving approach heavily depends on

- the heuristic to select the variable X_i,
- the strategy to split the domain D_i into $\grave{D}_i$ and $\acute{D}_i$, and
- the efficiency of the propagation methods to prune the domains of the CSP.

A common strategy is to select the variable with the smallest domain with size > 1, since it seems most constrained, and to split the domain into a single valued domain $\grave{D}_i$ and the remaining values ($\acute{D}_i = D_i \setminus \grave{D}_i$). This heuristic enables a strong reasoning within the subproblem containing $\grave{D}_i$, which enables an early failure detecting. Therefore, it is known as the *first-fail* strategy. The value selection for $\grave{D}_i$ is still problem specific.

Constraint Satisfaction Problems are used to find and enumerate solutions of a problem. If one has to solve a problem that involves the minimization/maximization of some objective function f, CSP formulations without further extensions are usually inefficient since they would require the evaluation of all possible solutions. Instead, a *Constraint Optimization Problem (COP)* can be formulated that joins a CSP with an objective function f. Thus, a *finite domain COP* is defined by a quadruple $\langle \mathcal{X}, \mathcal{D}, \mathcal{C}, f \rangle$, where

- $\mathcal{X} = \{X_1, \ldots, X_n\}$ is the finite set of n variables describing the problem,
- $\mathcal{D} = \{D_1, \ldots, D_n\}$ is a set of finite domains, where D_i describes the domain of X_i,
- $\mathcal{C}$ is a set of constraints defining the relations between the variables in $\mathcal{X}$, and
- $f : D_1 \times \ldots \times D_n \to \mathbb{R}$ is the objective function to be optimized.

In the following, w.l.o.g., we assume the objective function f to be minimized. A solution A_s of a COP is a solution of the CSP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ that minimizes $f(A_s(X_1), \ldots, A_s(X_n))$, written shortly $f(A_s)$. Thus, for all other solution A'_s of the CSP it holds $f(A_s) \leq f(A'_s)$.

To solve COPs, usually *branch-and-bound* methods are applied. As a rough description, they follow a first-fail strategy as for CSPs but ignore the majority of subproblems that can not contain the optimal solution according to f. The pruning of subproblems $\langle \mathcal{X}, \mathcal{D}', \mathcal{C}' \rangle$ is based on the calculation of a lower bound $b_f(\mathcal{D}')$ for f based on the current domains $\mathcal{D}'$, even if not assigned. The lower bound b_f estimates the minimal value of f that can be reached by the subproblem. Thus, if the lower bound $b_f(\mathcal{D}')$ exceeds the objective reached so far, the current subproblem is ignored.

In short, the branch-and-bound method can be sketched as follows: Given a COP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C}, f \rangle$, we calculate a solution A_0 of the CSP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$. If it does not exist, the COP also has no solution. If A_0 exists, we derive a minimal lower bound $b_{\min} = f(A_0)$ that has to be preserved by any following solution, i.e. in all following subproblems $\langle \mathcal{X}, \mathcal{D}', \mathcal{C}' \rangle$ we enforce the constraint $b_f(\mathcal{D}') < b_{\min}$ and add it to $\mathcal{C}'$. In consequence, we get only solutions that are better with respect to f. If we can find a solution A_1 for the altered subproblem, we update the minimal lower bound $b_{\min} = f(A_1)$ and continue the search until no solution (falling below the current lower bound) can be found. Thus, the last solution found is the overall solution of the COP.

The branch-and-bound protocol works most efficiently if the lower bound calculation $b_f(\mathcal{D}')$ is close to the best value of f that can be achieved within $\langle \mathcal{X}, \mathcal{D}', \mathcal{C}' \rangle$. Only then it is possible to do an extensive pruning of subproblems, resulting in reduced computation times.

1.5 Thesis Structure

As already discussed, this thesis studies lattice proteins that are a common abstraction to assess the structure and sequence space of proteins on a level that is computationally accessible for large scale studies but still expressive enough to enable relevant conclusions. Within this Chapter I, the background information, definitions, and general methods were given that are needed in the rest of this thesis. Furthermore, a brief introduction of the major fields covered by this manuscript was given. These are (1) the generation of models for real protein structures, (2) optimal structure prediction, (3) folding simulation, and (4) sequence design that enables evolutionary studies.

This thesis covers a wide spectrum of approaches and applications for all four problems and therefore is split into according chapters. Each of the chapters comes with a brief introduction of the problem studied, summarizes related work from literature, and motivates the work presented within. Due to the wide range of different problems, each chapter is completed with a summary and discussion of the findings. The chapters follow the order of the introduced fields. Chapter VII summarizes the presented and envisions future work.

The whole manuscript is written in a "we" rather than an "I" form. I decided on that since the majority of the work is based on team work with various collaboration partners that accompanied me. Most of the ideas are spawned in discussions and brain stormings that have been a well of inspiration and a central motivation for all my work. Consequently, the "we" form is used in the following.

CHAPTER II

Lattice Models of Real Protein Structures

2.1 Introduction

It is not always computationally feasible to undertake protein structure studies using full atom representations. The challenge is to reduce complexity while maintaining detail. Lattice protein models are often used to achieve this and generally only the protein backbone or the amino acid centre of mass is represented (*Mirny and Shakhnovich*, 2001; *Dill et al.*, 2008; *Istrail and Lam*, 2009).

In order to evaluate the applicability of different lattices and to enable the transformation of real protein structures into lattice models, a representative lattice protein structure has to be calculated. The quality of the representative is assessed by some structural distance measure that compares the original and the model. Figure 2.1 depicts the general scheme for solving the problem.

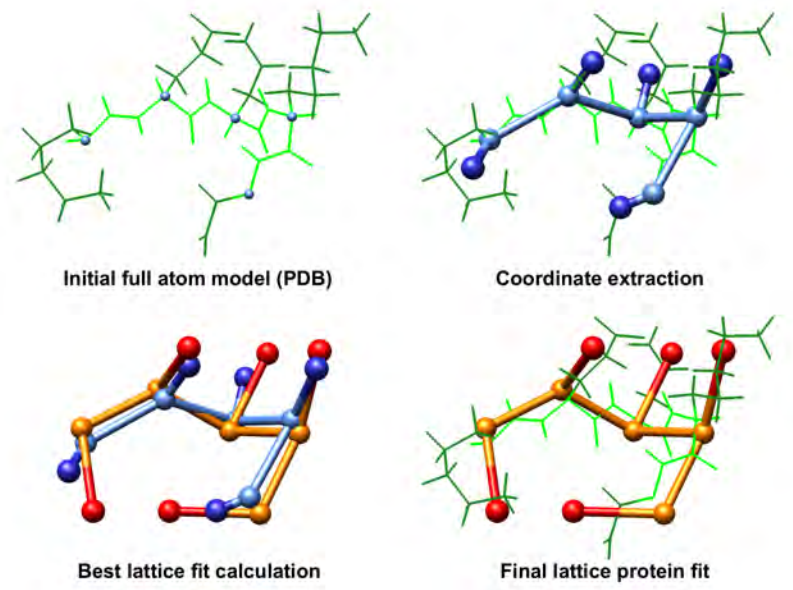

Initial full atom model (PDB)
Coordinate extraction
Best lattice fit calculation
Final lattice protein fit

Figure 2.1: The Protein Chain Lattice Fitting (PCLF) problem is to calculate the optimal lattice protein model (orange/red) of a protein (light-/darkgreen) given in full atom representation. It requires the minimization of a structural distance measure between the extracted protein coordinates to fit (cyan/blue) and the model.

Maňuch and Gaur (2008) have shown the NP-completeness of this problem for backbone-only models in the 3D-cubic lattice and named it the *protein chain lattice fitting (PCLF) problem.* The most important aspects in producing lattice protein models with a low structural distance to the original protein are the lattice coordination number and the neighbourhood vector angles (*Park and Levitt*, 1995; *Pierri et al.*, 2008). Lattices with intermediate coordination numbers, such as the face-centered-cubic (FCC) lattice, can produce high resolution backbone models (*Park and Levitt*, 1995).

Plan of the Chapter

Within this chapter, a new approaches to tackle the PCLF problem are introduced.

Before presenting the approaches, formal definitions of the PCLF problem and the applied structural distances are given within this section. They are followed by an overview over existing approaches in the field.

Afterwards, we present in Sec. 2.2 our LatFit tool. It implements a chain growth procedure that minimizes the intramoleculare distance deviations between original and model structure (measured via dRMSD). LatFit can be used to calculate backbone-only as well as side chain models in arbitrary lattices. Since we use a heuristic that produces approximate solutions, it might be possible to refine the models obtained.

To tackle this problem, we introduce in Sec. 2.3 a constraint-based refinement strategy that enables a further improvement of lattice protein models for a given structure. The refinement protocol reduces the structural bias resulting from heuristic approaches. We investigate and evaluate a global and and stepwise refinement protocol where the latter can be applied within local search to find the globally best solution.

2.1.1 Problem Definition

In the following the PCLF problem is introduced for backbone-only models, which is easily extended and alike formulated to side chain models.

Given a protein in full atom representation of length n (e.g. in Protein Data Base (PDB) format (*Berman et al.*, 2000)), we denote the sequence of 3D-coordinates of its C_α-atoms (its backbone trace) by $O = (O_1, \ldots, O_n)$. These coordinates have to be represented by the lattice protein model.

The protein model has to be placed into a lattice L defined by its neighboring vectors N_L (see Sec. 1.2.1). All neighboring vectors $\vec{v} \in N_L$ of the used lattice L are scaled to a length of $3.8\mathring{A}$, which is the mean distance between consecutive C_α-atoms in real protein structures (*Park and Levitt*, 1995).

Within L, a backbone-only model $P = (P_1, \ldots, P_n)$ of O is a sequence of n lattice nodes that are linearly connected and self-avoiding (see Sec. 1.2.2).

The *PCLF problem* is to find a lattice protein model P of a given protein's backbone O, such that a distance measure $\text{dist}(P, O) : L^n \times L^n \to \mathbb{R}$ between P and O is minimized (*Maňuch and Gaur*, 2008). i.e.

$$\forall_{P'} : \text{dist}(P, O) \leq \text{dist}(P', O) \ . \tag{2.1}$$

2.1.2 Structural Distance Measures

The PCLF problem is to minimize the structural distance between the original protein to be fitted and the derived model. Within the literature, different distance measures have been used. The most prominent measures used both in the literature and in this thesis are presented here:

2.1.2.1 Root Mean Square Deviation - RMSD

The *Root Mean Square Deviation (RMSD)* is a measure frequently used to evaluate differences between structural data. Based on individual differences between corresponding data points, the *residuals*, an overall measure is aggregated. For the comparison of two protein structures, two types of RMSDs have been introduced in the literature.

Given two protein structures P and O, the *coordinate RMSD (cRMSD)* evaluates the intermolecular pairwise distances (squared error) between corresponding protein monomer coordinates P_i, O_i, see Eq. 2.2 and 2.3, where $|\vec{x}|$ denotes the vector length of $\vec{x}$. Thus, the cRMSD can be calculated in linear time ($\mathcal{O}(n)$) but depends on the superposition of the two structures, i.e. it is sensitive to translation and rotation. To obtain a superpositioning before calculating the final cRMSD the algorithm by *Kabsch* (1976, 1978) can be used. It applies linear and vector algebra in order find the optimal translation and rotation for the two structures and thus minimizes the according cRMSD.

$$\begin{aligned} &\underline{\text{backbone-only models}} && P = (P_1, \ldots, P_n) \text{ and } O = (O_1, \ldots, O_n) \\ &\text{cRMSD}(P, O) \quad = && \sqrt{\frac{\sum_i |P_i - O_i|^2}{n}} \end{aligned} \tag{2.2}$$

$$\begin{aligned} &\underline{\text{side chain models}} && P = ((P^b_1, P^s_1), \ldots, (P^b_n, P^s_n)) \text{ and } O = ((O^b_1, O^s_1), \ldots, (O^b_n, O^s_n)) \\ &\text{cRMSD}(P, O) \quad = && \sqrt{\frac{\sum_i |P^b_i - O^b_i|^2 + \sum_i |P^s_i - O^s_i|^2}{2n}} \end{aligned} \tag{2.3}$$

In contrast, the *distance RMSD (dRMSD)* is a superposition-independent measure. It evaluates all intramolecular pairwise distances within P and their equivalents in O, see Eq. 2.4 and 2.5. By that, dRMSD is computationally more expensive ($\mathcal{O}(n^2)$) and insensitive to reflections.

$$\begin{aligned} &\underline{\text{backbone-only models}} && P = (P_1, \ldots, P_n) \text{ and } O = (O_1, \ldots, O_n) \\ &\text{dRMSD}(P, O) \quad = && \sqrt{\frac{\sum_{i<j} (|P_i - P_j| - |O_i - O_j|)^2}{(n^2 - n)/2}} \end{aligned} \tag{2.4}$$

$$\underline{\text{side chain models}} \qquad P = ((P_1^b, P_1^s), \ldots, (P_n^b, P_n^s)) \text{ and } O = ((O_1^b, O_1^s), \ldots, (O_n^b, O_n^s))$$

$$\mathrm{dRMSD}(P, O) = \sqrt{\frac{\begin{array}{rll} & \sum_{i<j}(|P_i^b - P_j^b| - |O_i^b - O_j^b|)^2 & \text{backbone} \\ + & \sum_{i<j}(|P_i^s - P_j^s| - |O_i^s - O_j^s|)^2 & \text{side chain} \\ + & \sum_{i \neq j}(|P_i^b - P_j^s| - |O_i^b - O_j^s|)^2 & \text{backbone / side chain} \\ + & \sum_{i}(|P_i^b - P_i^s| - |O_i^b - O_i^s|)^2 & \end{array}}{2n^2 - n}} \tag{2.5}$$

2.1.2.2 Global Distance Test - GDT

The *Global Distance Test (GDT)* measures the structural diversity of two structures. It yields a more accurate measurement than the more common RMSD, which is sensitive to outlier regions created by poor modeling of individual regions in a structure that is otherwise reasonably accurate. GDT is used as a major criteria within the Critical Assessment of Structure Prediction (CASP) competition (*Kryshtafovych et al.*, 2009) where current approaches and methods are evaluated and compared (*CASP*, 2010).

The GDT score is defined by the largest set of amino acids where the C_α carbon atoms in the model structure $P = (P_1, \ldots, P_n)$ fall within a defined distance cutoff $\mathrm{GDT}_{\max}$ of their corresponding positions in the experimental structure $O = (O_1, \ldots, O_n)$. The GDT measure can be computed in linear time ($\mathcal{O}(n)$) but is superposition-dependent due to coordinate comparison. The GDT score can be computed under several cutoff distances. The GDT score generally increases with increasing cutoff, since a larger cutoff contains the covered cases of a smaller one. Based on the applied cutoffs one distinguishes between the *high accuracy GDT (GDT_HA)* from Eq. 2.6 and the more general *total score GDT (GDT_TS)* given in Eq. 2.7 (*CASP8*, 2009).

$$\text{GDT\_HA(P,O)} = \frac{\Delta(0.5\mathring{A}) + \Delta(1\mathring{A}) + \Delta(2\mathring{A}) + \Delta(4\mathring{A})}{4} \tag{2.6}$$

$$\text{GDT\_TS(P,O)} = \frac{\Delta(1\mathring{A}) + \Delta(2\mathring{A}) + \Delta(4\mathring{A}) + \Delta(8\mathring{A})}{4} \tag{2.7}$$

$$\text{with}$$

$$\Delta(\mathrm{GDT}_{\max}) = \frac{|\{\ P_i \ \mid \ \text{where } |P_i - O_i| \leq \mathrm{GDT}_{\max}\}|}{n}$$

2.1.3 Existing Approaches

The PCLF problem has been widely studied for backbone-only models. Suggested approaches use quite different methods.

To calculate the optimal fit, *Covell and Jernigan* (1990) applied full enumeration and evaluation of all possible lattice protein structures. To reduce the computational cost, *Hinds and Levitt* (1992) developed a dynamic programming scheme for the optimization. Due to the enormous number of possible structures, both approaches are not feasible for longer sequences.

For the calculation of nearly optimal, approximate fits heuristics have been developed. They range from greedy chain growth strategies (*Park and Levitt*, 1995; *Miao et al.*, 2004) via simulated annealing protocols (*Ponty et al.*, 2008) to the optimization of specialized force fields (*Koehl and Delarue*, 1998; *Reva et al.*, 1998).

For side chain lattice models only one approach by *Reva et al.* (1995) has been published.

2.2 Heuristic Structure Fitting via LatFit

In contrast to the large number of published methods to derive lattice protein models from real proteins few, if any, are publicly available or directly applicable. To our knowledge, there is only one publicly available implementation, namely `LocalMove` by *Ponty et al.* (2008), to derive lattice protein models from real proteins. `LocalMove` was introduced for backbone-only models in 3D-cubic and FCC lattice.

In the following, the tool LatFit is presented, which is freely available within the LatPack-package (*Mann et al.*, 2008a). LatFit solves the PCLF problem, i.e. transforms a protein from full atom coordinate data into a lattice model, and is available as both a stand-alone tool for high-throughput pipelines and a web interface for *ad hoc* usage. A new fitting procedure that optimizes the distance RMSD enables rotation independent lattice model creation of protein structures. The method implements a chain-growth procedure and is applicable to arbitrary lattices. Both backbone and side chain representations are calculated with equivalent accuracy. A depiction of the web interface is given in Fig. 2.2.

Using LatFit we present the first comprehensive study of lattice quality for protein models including side chains. In our test, LatFit fitted the majority of models on an FCC lattice within $1.5\mathring{A}$ RMSD.

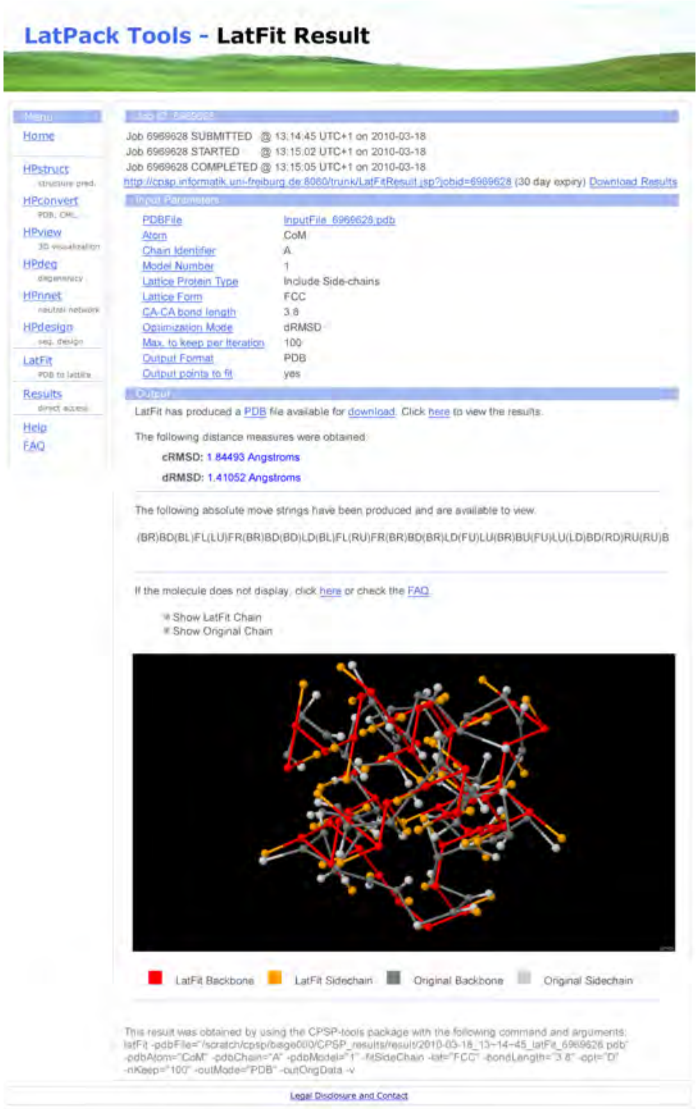
2.2.1 Method

First, all neighboring vectors $\vec{v} \in N_L$ of the used lattice L are scaled to a length of $3.8\mathring{A}$, which is the mean distance between consecutive C_α atoms and close to the mean distance between a C_α atom and the associated side chain centroid ($\approx 3.6\mathring{A}$). This scaling enables a reasonable mapping of the protein onto the lattice: connected monomers from the backbone will show the desired mean distance ($|P_{i+1} - P_i| = |\vec{v}| = 3.8\mathring{A}$) while in side chain models the distance distortion from backbone to side chain monomers is small (using $|P^s_i - P^b_i| = |\vec{v}| = 3.8\mathring{A}$ instead of $3.6\mathring{A}$). The main advantage is that all resulting measures will be directly interpretable in $\mathring{A}$ units.

The positions for each amino acid i to be fitted, i.e. the C_α position of the backbone O^b_i, and the centroid O^s_i (geometric center) of all non-hydrogen atom coordinates of the side chain, are extracted from the PDB file. This results in the protein positions $O = ((O^b_1, O^s_1), \ldots, (O^b_n, O^s_n))$ to be modeled.

The fitting follows a greedy iterative chain-growth procedure. The initial lattice model's backbone and side chain position (P^b_1 and P^s_1) are placed arbitrarily but adjacent ($P^b_1 - P^s_1 \in N_L$). For each iteration $1 < i \leq n$, all valid placements of the next P^b_i and P^s_i on the lattice are calculated. A distance RMSD (dRMSD, Eq. 2.5) evaluation is used to identify the best n_{keep} structures of length i for the next extension iteration. Since dRMSD is a rotation/reflection independent measure, symmetric structures must be filtered. The iterative procedure results in a dRMSD optimized fit $P = ((P^b_1, P^s_1), \ldots, (P^b_n, P^s_n))$.

As discussed in Sec. 2.1.2.1, distance RMSD is blind to reflections and independent from translations and rotations. Thus, to calculate the final fit P' for the initial protein O, a superpositioning of the dRMSD-optimised structure P and a reflected version $\bar{P}$ is done using the method by *Kabsch* (1978). The superpositioning translates and rotates $P/\bar{P}$ in order to achieve the best mapping



Job 6969628 SUBMITTED @ 13:14:45 UTC+1 on 2010-03-18
Job 6969628 STARTED @ 13:15:02 UTC+1 on 2010-03-18
Job 6969628 COMPLETED @ 13:15:05 UTC+1 on 2010-03-18
http://cpsp.informatik.uni-freiburg.de:8080/trunk/LatFitResult.jsp?jobid=6969628 (30 day expiry) Download Results

PDBFile	InputFile_6969628.pdb
Atom	CoM
Chain Identifier	A
Model Number	1
Lattice Protein Type	Include Side-chains
Lattice Form	FCC
CA-CA bond length	3.8
Optimization Mode	dRMSD
Max. to keep per iteration	100
Output Format	PDB
Output points to fit	yes

LatFit has produced a PDB file available for download. Click here to view the results.

The following distance measures were obtained:

cRMSD: 1.84493 Angstroms

dRMSD: 1.41052 Angstroms

The following absolute move strings have been produced and are available to view:

(BR)BD(BL)FL(LU)FR(BR)BD(BD)LD(BL)FL(RU)FR(BR)BD(BR)LD(FU)LU(BR)BU(FU)LU(LD)BD(RD)RU(RU)B

If the molecule does not display, click here or check the FAQ.

☑ Show LatFit Chain
☑ Show Original Chain

🟥 LatFit Backbone 🟧 LatFit Sidechain ⬛ Original Backbone ⬜ Original Sidechain

This result was obtained by using the CPSP-tools package with the following command and arguments:
latFit -pdbFile="/scratch/cpsp/bage000/CPSP_results/result/2010-03-18_13-14-45_latFit_6969628.pdb" -pdbAtom="CoM" -pdbChain="A" -pdbModel="1" -fitSideChain -lat="FCC" -bondLength="3.8" -opt="D" -nKeep="100" -outMode="PDB" -outOrigData -v

Figure 2.2: The LatFit web interface is part of the CPSPweb-tools (*Mann et al.*, 2009c).

onto O. The superpositioning with lowest coordinate RMSD (cRMSD, Eqn. 2.3) is selected and finally returned as P'.

Algorithm 3 outlines the whole approach. For simplicity, it is shown for backbone-only models. Since the dRMSD is based on a simple sum of distances (see 2.4), no full dRMSD computation has to be done in line 11. It is sufficient to update the dRMSD of the elongated fit $(P_1, \ldots, P_{(i-1)})$ with the sum of distance differences of the appended monomer P_i to $P_1, \ldots, P_{(i-1)}$ compared to the original chain O. This reduces the time complexity for each dRMSD evaluation during the elongation from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. The restriction of B to the n_{keep}-best elongations in line 11 can be done iteratively by storing only the n_{keep}-best seen so far in line 7. Thus, the overall runtime of LatFit is within $\mathcal{O}(n_{keep} \cdot |N_L| \cdot n^2)$, because the superpositioning algorithm by *Kabsch* (1976, 1978) used in line 14 runs in $\mathcal{O}(n)$.

Algorithm 3 LatFit for backbone-only lattice protein models

```
$B \leftarrow \{(P_1)\}$    ▷ best structure fits of last iteration
                            ▷ initialized with $(P_1)$, i.e. a single monomer structure representing $O_1$
$C \leftarrow \emptyset$    ▷ structures generated in current iteration
for $i = 2 \ldots n$ do
    for all $P = (P_1, \ldots, P_{(i-1)}) \in B$ do    ▷ $P$ has length $(i-1)$
        for all $\vec{v} \in N_L$ do
            if $(P_{(i-1)} + \vec{v}) \notin \{P_1, \ldots, P_{(i-1)}\}$ then    ▷ check self-avoidingness
                $C \leftarrow C \cup \{(P_1, \ldots, P_{(i-1)}, P_{(i-1)} + \vec{v})\}$    ▷ store elongation
            end if
        end for
    end for
    $B \leftarrow$ best $n_{keep}$ fits of $C$ according to dRMSD to $(O_1, \ldots, O_i)$
    $C \leftarrow \emptyset$    ▷ reset structure storage
end for
report best superposition fit $P \in B$ according to cRMSD compared to $O$
```

2.2.2 Results

We use LatFit to derive protein models on the commonly used 3D cubic (CUB), face-centered-cubic (FCC), and knights walk (210) lattices. Our test set was taken from the PISCES web server (*Wang and Dunbrack*, 2005) on March 10^{th} 2009 (40% sequence identity cut-off, chain length 50-300, R-value[1] ≤ 0.3 and resolution $\leq 1.5\mathring{A}$). Given our requirement for side chains, C_α-only chains were ignored. The resulting benchmark set contains 1198 proteins exhibiting a mean length of 160 ($\sigma = 64$). The dRMSD optimisation was parametrized with $n_{keep} = 1000$.

In accordance with previous studies (*Park and Levitt*, 1995; *Ponty et al.*, 2008; *Pierri et al.*, 2008), cRMSD and dRMSD are used to assess model quality. cRMSD measures the similarity in coordinate positioning whereas dRMSD measures the similarity of intramolecular distances (see Sec. 2.1.2.1). Due to the scaling of our lattice, RMSD results are in $\mathring{A}$ rather than the scaled values provided by *Ponty et al.* (2008) or *Pierri et al.* (2008).

Our backbone-only model RMSD values presented in Tab. 2.1 are competitive or superior to known fitting results from the literature (*Park and Levitt*, 1995; *Godzik et al.*, 1993; *Miao et al.*, 2004) and reproduce the high quality previously achieved by other methods using the FCC and 210 lattices.

LatFit is designed for side chain models and, as shown in Tab. 2.2, it performs very well. In general, side chain models produce slightly larger RMSD values than the equivalent backbone model. This is due to the fact that the variation in distance between consecutive C_α atoms (fitted in both models) is lower than that between C_α atoms and their side chain centroid (fitted only in side chain models). In our lattice model, every distance is fixed to $3.8\mathring{A}$ which results in a higher mean displacement of the side chain. Furthermore, even the amino acid glycine, that does not possess a side chain, is represented by 2 monomers in the lattice model that have to be fitted (both onto the original C_α position). Nevertheless, high accuracy fits are still attained. Results in our test set have mean dRMSDs of about 1.5 and $1.2\mathring{A}$ in the FCC and 210 lattice respectively. The strength of LatFit is its ability to produce both side chain and backbone-only lattice protein models. High

[1] The R-value measures the quality of the atomic structure model via a comparison to simulated data and is provided within the PDB protein database. A totally random model will give an R-value of about 0.63, whereas a perfect model will produce a value of 0.

	Results taken from *Park and Levitt* (1995)		Results taken from *Ponty et al.* (2008)	LatFit backbone-only models	
	dRMSD	cRMSD	cRMSD	dRMSD	cRMSD
	μ	μ	μ (rescaled to $\mathring{A}$)	μ / σ	μ / σ
CUB	2.34	2.84	3.5 ($0.923 \cdot 3.8$)	2.042 / 0.228	2.539 / 0.234
FCC	1.46	1.78	-	1.319 / 0.086	1.641 / 0.090
210	1.02	1.24	-	0.931 / 0.060	1.154 / 0.060

Table 2.1: Comparison of the RMSD mean values (μ) and standard deviations (σ) from literature to the results from our LatFit method for *backbone-only models* on three different 3D lattices (CUB - cubic, FCC - face-centered-cubic, 210 - knights walk).

	LatFit - side chain models	
	dRMSD	cRMSD
	μ / σ	μ / σ
CUB	2.779 / 0.754	4.157 / 1.331
FCC	1.496 / 0.153	2.104 / 0.246
210	1.126 / 0.068	1.601 / 0.100

Table 2.2: The RMSD mean values (μ) and standard deviations (σ) of the LatFit results for *side chain models* on three different 3D lattices (CUB - cubic, FCC - face-centered-cubic, 210 - knights walk).

accuracy models can be produced on the FCC lattice in seconds to minutes. Fits on the 210 lattice take orders of magnitude longer for relatively little gain in model accuracy. The reason is the vast size of the structure space as shown in Sec. 1.2.2. For this reason we recommend to use the FCC lattice for detailed high-throughput protein structure studies in both backbone-only and side chain representing lattice models.

2.2.3 Further Features

Coordinate data in the PDB is often incomplete. For example flexible loop structures are hard to resolve by current methods (*Choi and Deane*, 2010). This results in missing coordinate data for certain substructures within PDB files. LatFit enables a structural fitting of even such fragmented PDB structures. It produces a lattice protein fragment for each fragment of the original protein while ensuring that all are placed in the same lattice orientation and correct relative placement.

Currently, LatFit supports the 2D-square, 3D-cubic (CUB,100), 3D face-centered-cubic (FCC, 110) and 3D knights walk (210) lattice. The modular software design of our open source program enables an easy and straight forward implementation of other lattices via a specification of the according neighboring vectors N_L.

Supported output formats of LatFit are the PDB format, the Chemical Markup Language (CML) format, as well as a simple XYZ coordinate output. A highly compact string representation of the lattice protein is also given in absolute move strings that encode the series of neighboring vectors $\vec{v} \in N_L$ along the structure (see Sec. 1.2.2.3).

The generated absolute move string can be directly used to apply other lattice protein tools onto the resulting structures, e.g. from the CPSP-package for HP-type lattice protein models (*Mann et al.*, 2008b, 2009c) or from the LatPack tools for arbitrary lattice models (*Mann et al.*, 2008a, 2009b).

The web interface of LatFit, integrated into the CPSP-web-tools (*Mann et al.*, 2009c), enables *ad hoc* usage of the tool. Either a protein structure in PDB format can be uploaded or a valid identifier from the PDB database given. In the latter case, the according protein data is automatically retrieved from the protein data base.

Our default parameters enable a direct application of LatFit resulting in a balanced trade off between runtime and fitting quality. The computations are done remotely on a computation cluster while the user can trace the processing status via the provided job identifier and according link. Results are available and stored for 30 days.

Results can be visualised using `Jmol` (*Herráez*, 2006) for an interactive presentation of the final protein structure. The output file is also available for download. The final dRMSD and cRMSD values of the lattice protein compared to the original protein are given in concert with the absolute move string encoding of the resulting structure. For an example of the LatFit web interface see Fig. 2.2.

Further details regarding the methods implemented, the output formats supported and the applicable parametrisation are located in the LatFit manual distributed with the source code. We provide an extensive help page and a frequently asked questions (FAQ) section within the web interface at

`http://cpsp.informatik.uni-freiburg.de`

The source code of the stand-alone tools is available at

`http://www.bioinf.uni-freiburg.de/Software/LatPack/`

2.3 Constraint-based Refinement of Lattice Models

Publication

Lattice model refinement of protein structures
M. Mann *and A. Dal Palù*
in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'10)*, 2010
arXiv:1005.1853

Most of the PCLF methods introduced in literature are heuristics to derive good solutions in reasonable time (see Sec. 2.1.3). Greedy methods, as the chain growth algorithms by *Miao et al.* (2004) and *Park and Levitt* (1995) or the LatFit tool introduced in Sec. 2.2, enable low runtimes but the fitting quality depends on the chain growth direction and parametrization. Thus, resulting lattice models are biased by the method applied and have potential for refinement.

This section has the goal to provide some evidence that greedy methods can be effectively improved by subsequent refinement steps. We present a Constraint Programming based formalization using Limited Discrepancy Search and a simple working prototype (*Mann and Palu*, 2010).

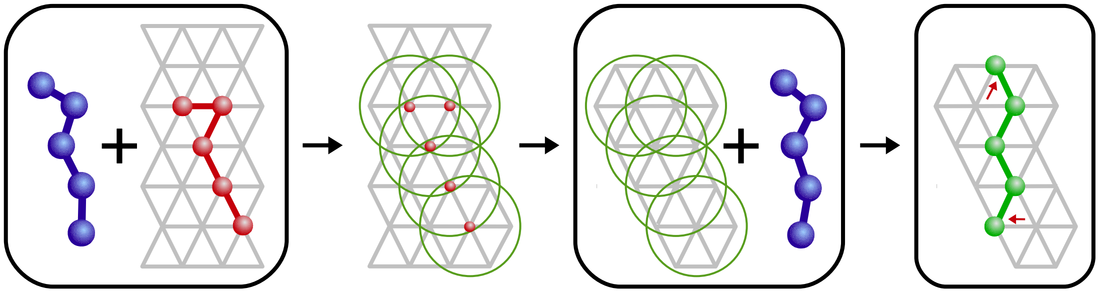


Figure 2.3: Workflow of the constraint-based refinement protocol. From left to right: (1, left) The protein to fit O (blue) and its initial lattice protein model P (red) in a 2D-triangular lattice (gray) are given. (2) For each monomer position of the inital model (in red), lattice points within a given radius (here $d_{\max} = 1$) are identified (green circles). (3) Based on these domains (circles) the COP for the refined protein model P' is defined that minimizes the objective function for the given original protein to fit (in blue). (4, right) The solution of the COP, i.e. a refined lattice protein model P' (green), that is placed within the relaxed domains. Red arrows indicate the moved monomer positions of the refined model compared to the initial one.

In the following, we tackle the *PCLF refinement problem* exemplified for backbone-only models. Here, a protein backbone $O = (O_1, \ldots, O_n)$ as well as a first lattice model $P = (P_1, \ldots, P_n)$ of O is given, e.g. derived by a greedy chain growth procedure like LatFit or the methods from (*Miao et al.*, 2004; *Park and Levitt*, 1995). The problem is to find a lattice model P' via a restricted relaxation/refinement of the original model P, such that its structural distance is smaller, i.e. $\text{dist}(P', O) < \text{dist}(P, O)$.

In the following, we use distance RMSD (dRMSD, Eq. 2.4) as the distance measure $\text{dist}(P, O)$. dRMSD is independent of the relative orientation of P and O since it captures the model's deviation from the intramolecular pairwise distances of C_α-atoms in the original protein. Minimizing this measure optimizes the lattice model obtained.

2.3.1 Approaches

In this section we formalize Constraint Programming based approaches to solve the PCLF refinement problem. We first formulate a Constraint Optimization Problem and show how to implement it in COLA, a COnstraint solver for LAttices (*Dal Palu et al.*, 2004). This is followed by an altered formulation that utilizes Limited Discrepancy Search (*Harvey and Ginsberg*, 1995).

2.3.1.1 Constraint Optimization Problem

As a first and straightforward approach we formulate a Constraint Optimization Problem (COP, see Sec. 1.4.3) for a given protein structure to fit $O = (O_1, \ldots, O_n)$ and its initial lattice protein model $P = (P_1, \ldots, P_n)$ within lattice L. We allow each position P_i of the initial model to be shifted within a radius of $d_{\max}$ lattice units. In the end, we want to identify the best final model P' within this relaxation such that $\text{dRMSD}(P', O) < \text{dRMSD}(P, O)$ and it holds for all other relaxations $\check{P}$ that $\text{dRMSD}(P', O) \leq \text{dRMSD}(\check{P}, O)$.

The COP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C}, f \rangle$ for this problem can be formalized as follows:

$\mathcal{X} = (X_1, \ldots, X_n)$ variables representing $P' = (P'_1, \ldots, P'_n)$

$\mathcal{D} = (D_1, \ldots, D_n)$ variable domains with $D_i = \{\vec{v} \mid \vec{v} \in L \wedge |\vec{v} - P_i| \leq d_{\max} \cdot 3.8\}$,
i.e. a P_i surrounding sphere with lattice unit radius $d_{\max}$
scaled by $3.8\mathring{A}$, i.e. the length $|\vec{n}|$ of neighboring vectors $\vec{n} \in N_L$

$\mathcal{C} = \{SAW(X_1, \ldots, X_n)\}$ a self-avoiding walk constraint, e.g. split into a chain of binary `contiguous`(X_i, X_{i+1}) constraints ensuring $(X_i - X_{i+1}) \in N_L$ and a global `alldifferent`$(\mathcal{X})$ constraint enforcing $\forall_{i \neq j} X_i \neq X_j$

$f \equiv$ dRMSD objective function that implements the dRMSD
$= \sum_{i<j}(|X_i - X_j| - |O_i - O_j|)^2$ to be minimized

Thus, the domains for $d_{\max} = 0$ only contain the original lattice point P_i (domain size 1), while $d_{\max} = 1$ results in P_i as well as all neighbored lattice points. For instance in the FCC lattice, the domain size is $1 + 12 = 13$ (see Sec. 1.2.1). The domain size guided by $d_{\max}$ defines the allowed relaxation of the original lattice model P to be refined. For more details about global constraints for protein structures on lattices, the reader can refer to (*Backofen and Will*, 2006; *Palù et al.*, 2010).

The COLA implementation takes advantage of the availability of 3D lattice point domains and distance constraints. The implementation adapts the original framework only in the input data handling and objective function definition. A working copy of COLA and the COP implemented is available at

`http://www2.unipr.it/~dalpalu/COLA/`

2.3.1.2 Limited Discrepancy Search

A simple enumeration with $d_{\max} = 1$ and a protein of length 50 in the FCC lattice already shows that the search space of the COP from the previous section is not manageable. In this example, each point can be placed in 13 different positions in the FCC lattice, and even if the `contiguous` constraint among the amino acids is enforced, the number of different structures to be considered is still beyond the current computational limits.

Since working with a COP, we tried a simple branch-and-bound search on $X_1, \ldots, X_n$ (see Sec. 1.4.3), where the dRMSD bound is estimated by considering all minimal pairwise distance contributions between monomers $X_i \neq X_j$. When the domains for an amino acid pair are not yet assigned it would provide a range of different contributions to the dRMSD measure. In order to estimate the minimal contribution, a closed formula computation based on bounding boxes of domains (rather than a full enumeration of all combinations) is assigned, see (*Palù et al.*, 2005) for further details. Clearly, this estimation is not particularly well-suited, since we get wrong minimal bounds for sparse domains that are wide spread or that have overlapping bounds. Unfortunately, the precise version requires a more expensive evaluation that boils down to full pair checks. Therefore, the current bound is very loose and the pruning effects are modest.

Protein ID	8RXN	1CKA	2FCW
length	52	57	106

Table 2.3: Used proteins from the Protein Data Base (PDB) (*Berman et al.*, 2000).

A general impression is that the dRMSD measure presents a pathological distribution of local minima, depending on the placement of amino acids on the lattice. In general, due to the discrete nature of the lattice, the modification of a single amino acid's position can drastically vary its contributions to the measure.

These considerations suggested to focus on the identification of solutions that improve the dRMSD w.r.t. P rather than searching for the optimal one. In terms of approximated search we tried to capture the main characteristics of the COP to design efficient and effective heuristics.

A simple idea we tested is the *limited discrepancy search* introduced by *Harvey and Ginsberg* (1995). Within this search, the amino acid placements in the current lattice model P' are compared to the initial one P. Every time a corresponding amino acid is placed differently in the two conformations, we say that there is a *discrepancy.* We set a global constraint that limits the number of such discrepancies to at most K. This allows to generate conformations that are rather similar to P, especially if $d_{\max}$ is greater than 1. The count of the number of discrepancies K is implemented directly in COLA at each search/assignment step.

The rational behind this heuristics is that we expect that potential conformations P' improve the dRMSD only when contained in a close neighborhood of the initial model structure P. The identification of locally optimized structures P' opens the setup of an iterative refinement protocol using a local search strategy. For instance we can apply a gradient walk scheme as introduced in Sec. 1.4.2.3 that follows the local gradient in dRMSD to a (local) minimum. But first we have to investigate if local changes via a K-limited discrepancy search can yield refined protein models.

2.3.2 Results

The initial lattice models to be refined were generated using LatFit introduced in Sec. 2.2 in its implementation from the LatPack package (*Mann et al.*, 2008a). LatFit was parametrized to consider the best $n_{keep} = 100$ structures from each elongation for further growth (for details refer to Sec. 2.2). All experiments are based on backbone-only models in the 3D FCC lattice.

We tested three proteins (Table 2.3) and for each of them we used the conformation obtained by LatFit as the initial protein model P. The coordinates O of the original protein to be fitted were taken from the LatFit output as well.

For all initial models, we applied a K-limited discrepancy search using our COLA implementation and different parametrizations. Table 2.4 reports the best dRMSD of our new models P' found depending on the allowed monomer displacement guided by $d_{\max}$ and the number K of amino acids allowed to be placed differently from the initial model. Furthermore, time consumption for each parametrization is given. Note that if either $K = 0$ or $d_{\max} = 0$ only the input structure resulting from the greedy LatFit run can be generated.

These results, yet preliminary, offer an interesting insight into the distribution of suboptimal solutions. It is interesting to note, e.g., that better solutions are found by allowing a rather large local neighborhood ($d_{\max}$ parameter) for a few amino acids. On the other side, it seems that few dRMSD(O, P')

		K			
8RXN		1	2	3	4
	1	1.2319	1.2172	1.1639	1.1189
$d_{\max}$	2	1.2319	1.1674	1.1596	1.0884
	3	1.2319	1.1674	1.1596	1.0884
		K			
1CKA		1	2	3	4
	1	1.2226	1.2226	1.2226	1.2226
$d_{\max}$	2	1.2026	1.1887	1.1887	1.1887
	3	1.2026	1.1887	1.1887	1.1887
		K			
2FCW		1	2	3	4
	1	1.1353	1.1324	1.1317	1.1309
$d_{\max}$	2	1.1321	1.1300	1.1254	1.1200
	3	1.1321	1.1300	1.1254	1.1200

time in seconds

		K			
8RXN		1	2	3	4
	1	0.112	0.790	2.365	20.70
$d_{\max}$	2	0.068	0.983	6.500	106.6
	3	0.106	0.499	7.399	124.0
		K			
1CKA		1	2	3	4
	1	0.402	0.615	3.442	39.27
$d_{\max}$	2	0.225	0.456	7.595	120.6
	3	0.421	0.616	8.573	140.2
		K			
2FCW		1	2	3	4
	1	0.118	1.997	49.99	1128
$d_{\max}$	2	0.294	7.192	341.8	14235
	3	0.332	8.129	394.5	16140

Table 2.4: Influence of the parameters $d_{\max}$ and K on discrepancy search measured in the resulting final structural distance of the orignal protein data and the model (dRMSD(O, P')) and the time consumption.

modifications (K) are sufficient to alter the input structure P and to obtain a better conformation P'.

In Figure 2.4 we exemplify the gain of model precision for the protein 8RNX. Only the relaxation of $K = 4$ monomers enables the structural change that leads to a dRMSD drop from 1.2469 down to 1.0884, an improvement of about 13%. A movement of less monomers would not enable such a drastic change. This depicts the potential of a local search scheme that iteratively applies a series of such structural changes.

Investigating the time consumption (Table 2.4) one can see that the runtime increases drastically with K which governs the search space size. The domain sizes implied by $d_{\max}$ do not show such a strong influence.

2.4 Discussion and Future Work

The Protein Chain Lattice Fitting problem is a hard computational problem and usually tackled by heuristic approaches. Such approaches aim at a balance between computational effort, i.e. time consumption, and the quality of the lattice protein model produced.

The used LatFit method introduced in Sec. 2.2 implements a dRMSD-optimizing chain growth heuristic. This enables a superposition-independent structure optimization and yields very good results compared to literature. For *ad hoc* usage, it is the best available program to calculate backbone-only models and the only available approach to derive lattice protein models including side chains. Since LatFit is a heuristic, it does not necessarily produce the best fit. The chain growth protocol might bias the fit to be more accurate on the structure's head compared to the tail due to restricted degrees of freedom.

We have shown that a constraint-based search for dRMSD-lowering structural changes can significantly improve the already good fits produced by LatFit. Small structural changes can yield

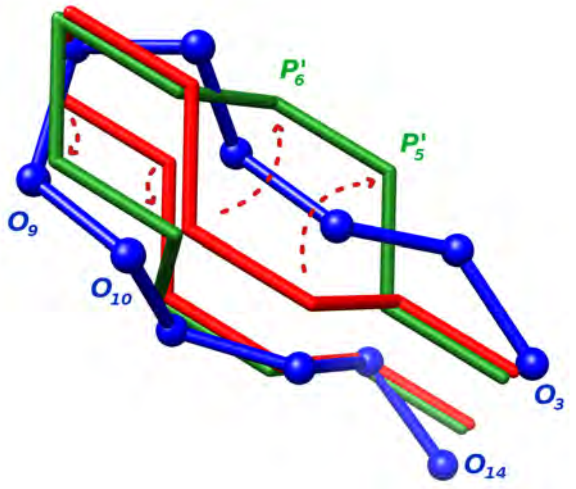


Figure 2.4: The altered loop region (3-14) of the initial lattice model P (red) for the original backbone O (blue, balls) of protein 8RNX together with the final/refined lattice model P' (green) resulting from parameterization $d_{\max} = 2$ and $K = 4$. Dashed red arrows indicate the shifted positions of the initial model at positions 5-6 and 9-10.

large enhancements while the computational costs within the Constraint Programming framework are small. The behavior encountered is an indicator that a local search exploring only limited structural changes should provide an efficient way to refine fitted lattice protein models. Furthermore, the generic formalization using a COP enables the extension of the refinement scheme to side chain models similar to the extension of local moves presented later in Chapter IV (*Mann et al.*, 2009b).

A hybrid approach of LatFit and the refinement protocol might therefore be well suited to yield close to optimal structural fits of protein structures. In a first optimization step, the LatFit chain growth procedure can be used to search for good initial model structures. These resulting structures are subsequently refined using the constraint-based refinement protocol combined with local search. Therein, the structure is enhanced as long as the structural distance can be reduced significantly. The final structure should be unbiased and close to optimal.

For the refinement implementation, we could follow *Ullah and Steinhöfel* (2010) who have extended COLA to be directly applicable within local search approaches. Thus, we can directly implement a lattice fit refinement tool. Another option is the reimplementation of the approach using the Constraint Programming system Gecode, the very efficient state-of-the-art CP system. This enables the use of a local neighboring search (*Cipriano et al.*, 2009; *Dotu et al.*, 2008) that allows to neatly integrate Local Search into the Gecode CP framework. The resulting program should outperform our prototypical implementation done in COLA.

CHAPTER III

Optimal Structure Prediction in HP-models

3.1 Introduction

The structure of a protein determines its function. Thus, the knowledge of a protein's three-dimensional structure enables its functional classification, the investigation of folding errors that cause dysfunction, or the study of the chemical processes catalyzed by the protein. Therefore, the determination of a protein's functional fold is a fundamental challenge in molecular biology and bioinformatics.

So far, protein structures have been determined by extensive experimental studies like X-ray crystallography, pioneered by *Kendrew et al.* (1953), or nuclear magnetic resonance (NMR) spectroscopy (*Wüthrich*, 1990). Both studies are complicated and expensive such that the number of known protein structures is magnitudes smaller than the number of known protein sequences. At the end of 2010, the Protein Data Bank (PDB (*Berman et al.*, 2000)) holds 70,184 structures[1]. In contrast, the National Center for Biotechnology Information (NCBI) sequence data base lists at the same time already more than 6.3 million genes for Bacteria and Eukaryota[2].

In order to increase the number of identified protein structures, many computational methods have been introduced. Every two years, the state-of-the-art algorithms are benchmarked within the CASP experiment (Critical Assessment of Techniques for Protein Structure Prediction, see (*Kryshtafovych et al.*, 2009)). At the moment, the overall accuracy of computational approaches is limited, but the quality of the prediction increases steadily.

Central to most approaches is the minimal energy assumption, i.e. that the functional fold corresponds to a minimal energy structure within the protein model (as already discussed in Sec. 1.1). Furthermore, conformational restrictions, e.g. of the torsion angles (*Ramachandran et al.*, 1963), or structural preferences are considered. Thus, a fundamental knowledge of the forces and principles of the structure formation process is essential to predict protein structures.

Simplified protein models, as lattice proteins, can be used to investigate such general principles since they allow for extensive studies with moderate computational effort. In the following, we investigate methods for the problem of energy minimal (optimal) structure prediction within the HP model (*Lau and Dill*, 1989).

Formally, given a lattice protein sequence S of length n and a lattice L, the problem is to determine a lattice model structure $P = (P_1, \ldots, P_n) \in L^n$ that minimizes the underlying energy

[1] Accessed on 1st April 2011 at `http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total`

[2] Accessed on 1st April 2011 at `http://www.ncbi.nlm.nih.gov/projects/Gene/gentrez_stats.cgi`

function E, i.e.$\forall_{\breve{P}\neq P}\ :\ E(S,P) \leq E(S,\breve{P})$. We can not enforce P to be the unique energy minimum for S since two structures can yield the same energy for the HP energy function defined in Sec. 1.2.3.1. This is known as the degeneracy problem and will be discussed within this chapter.

Despite the simplicity of the protein model, the so defined *protein structure prediction (PSP) problem* has been shown to be NP-complete in the 2D-square (*Crescenzi et al.*, 1998) as well as the 3D-cubic lattice (*Berger and Leighton*, 1998). Thus, simple schemes as exhaustive structure enumeration as done by *Irbäck and Sandelin* (2000) in 2D are not feasible or restricted to very short sequence lengths due to the fast growing structure space shown in Sec. 1.2.2.

Within the last decades, many approximative schemes have been introduced that allow for the determination of structures with an energy close to the optimum. For both, backbone-only (*Hart and Istrail*, 1996; *Newman*, 2002) and side chain models (*Hart and Istrail*, 1997; *Heun*, 2003), performance-guaranteed approximation algorithms have been developed. Since we are interested in exact solutions to the PSP problem we are not considering these approaches. For a detailed overview refer to recent reviews by *Hart and Newman* (2006) and *Istrail and Lam* (2009).

For the HP model, currently only two exact approaches to the PSP problem are known, namely the CHCC method by *Yue and Dill* (1995) and CPSP approach by *Backofen and Will* (2006), both are discussed in the following.

Given a protein sequence S with n_H hydrophobic residues, the *Constrained Hydrophobic Core Construction* (CHCC) algorithm by *Yue and Dill* (1995) enumerates all possible compact configurations of n_H H-monomers. The enumeration of these hydrophobic (H)-cores is done in decreasing compactness, i.e. decreasing number of possible HH-interactions. For each H-core, the CHCC method identifies all structures where H-monomers are bound to the current hydrophobic core's positions. The energy-decreasing testing enables the calculation of energy-optimal structures but it was shown by *Backofen and Will* (2006) that the CHCC method is not capable of calculating *all* optimal structures. In contrast, the *Constraint-based Protein Structure Prediction (*CPSP*)* approach by *Backofen and Will* (2006) was shown to be complete when calculating optimal structures.

Plan of the Chapter

In the following, we will briefly summarize the CPSP approach for backbone-only HP models introduced by *Backofen and Will* (2006). Based on that, we present in Sec. 3.2.2 our extension of the approach to side chain models in 3D lattices that is the first approach to predict optimal structures within these models. Both methods are available within our CPSP-tools package and its web front-end CPSP-web-tools summarized in Sec. 3.2.3. We apply our implementations in Sec. 3.3 for an extensive CPSP-based degeneracy study of the HP model. The observed immense number of optimal structures motivates the introduction of a hydrophobicity-based equivalence relation defition presented in Sec. 3.4 to group similar optimal structures. We introduce a new scheme to identify a representative optimal structure for each equivalence class. Such representative structures in the HP model can be used to improve local search for optimal solutions in more advanced energy models, e.g. the MJ-model. A possible hybrid approach is presented and evaluated at the end of the chapter in Sec. 3.5.

3.2 Constraint-based Protein Structure Prediction for Side Chain Models

Publications

- **CPSP-web-tools: a server for 3D lattice protein studies**
  ***M. Mann**, C. Smith, M. Rabbath, M. Edwards, S. Will, and R. Backofen*
  in *Bioinformatics*, vol. 25 (5), pp. 676–677, 2009
  doi:10.1093/bioinformatics/btp034

- **CPSP-tools - exact and complete algorithms for high-throughput 3D lattice protein studies**
  ***M. Mann**, S. Will, and R. Backofen*
  in *BMC Bioinformatics*, vol. 9, pp. 230, 2008
  doi:10.1186/1471-2105-9-230

The *Constraint-based Protein Structure Prediction (*CPSP*)* approach by *Backofen and Will* (2006) enables the calculation of a sequence's optimal structures without full structure space enumeration. Similar to the CHCC method, it utilizes the observation that optimal structures show a (nearly) optimal packing of H monomers, visualized in Fig. 3.1.

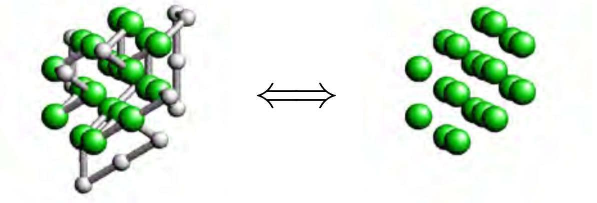


Figure 3.1: Coherence between an optimal structure (left) and the optimally dense packing of H-monomers, i.e. its H-core (right).

Due to the application of Constraint Programming, it results in a fast and flexible framework that can be easily extended. After reviewing the CPSP approach, we introduce it's extension to side chain models by altering the formulated Constraint Satisfaction Problems. All methods have been implemented and the resulting bundle of CPSP-tools and the according web interfaces are presented in Sec. 3.2.3.

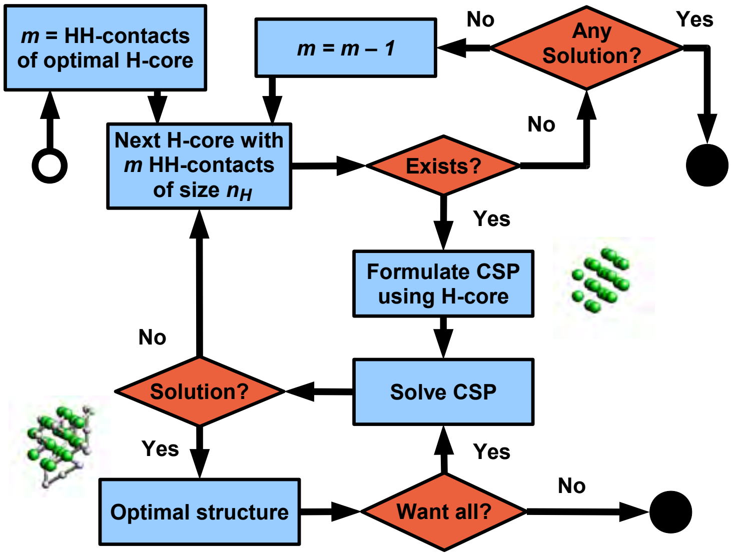

m = HH-contacts of optimal H-core
m = m – 1
No
Any Solution?
Yes
Next H-core with m HH-contacts of size n_H
Exists?
No
Yes
Formulate CSP using H-core
No
Solution?
Solve CSP
Yes
Yes
Optimal structure
Want all?
No

Figure 3.2: Workflow of the CPSP approach. The open circle marks the start while full circles represent program ends.

3.2.1 CPSP approach

The CPSP approach can be sketched in two major steps:

Step 1 : *H-core construction:* Given the number n_H of H monomers from the target sequence S, all optimal packings of n_H monomers are calculated. These *optimal H-cores* show the maximal number of contacts possible. For a fixed sequence S and the corresponding n_H, we denote the set of optimal H-cores with $\mathcal{H}$. The calculation of $\mathcal{H}$ is computationally difficult on its own and was solved by *Backofen and Will* (2001, 2006) using a combination of Dynamic Programming and Constraint Programming.

Step 2 : *Structure threading:* Given the protein sequence S and the according optimal H-cores $\mathcal{H}$, only structures are enumerated where H monomers are confined to an optimal H-core $h \in \mathcal{H}$. To this end, they are "threaded" through the H-cores such that each H-monomer is placed on an H-core position. Since all $h \in \mathcal{H}$ show the maximally possible number of contacts between H monomers, each resulting structure is optimal according to the HP energy function as well (see Sec. 1.2.3.1). The structure threading is done by solving a Constraint Satisfaction Problem (CSP) for each $h \in \mathcal{H}$ that is introduced in detail in Sec. 3.2.2.

Since step 1 depends only on the number of H monomers n_H and no further property of any sequence, we can precalculate the H-cores for different n_H and store them in a database. This significantly speeds up the approach and confines the computation time for a given sequence S to step 2 only, i.e. the structure threading.

It might happen, that we find no appropriate structure threading for a sequence S and the according set of optimal H-cores $\mathcal{H}$. Thus, we revert to the set of the best suboptimal H-cores $\mathcal{H}'$ that show at least one contact less than an optimal H-core $h \in \mathcal{H}$ and iterate the procedure. Still it holds: the first successive structure threading yields an optimal structure, since the the threading with compacter H-cores yielded no solution while the full list of such cores was tried. Figure 3.2 shows a graphical sketch of the whole CPSP method. For further details on the CPSP approach refer to the articles by *Backofen and Will* (2006) and the thesis of *Will* (2005).

When interested in all optimal structures, the CPSP approach can be directly used. First, one has to identify the first H-core where a solution can be found, i.e. an optimal structure can be threaded through. Afterwards, one enumerates all solutions for this H-core and for all H-cores that enable the same number of contacts. If the set of optimal H-cores is complete, the computed set of optimal structures will be complete too. The full enumeration scheme is sketched in Fig. 3.2.

The approach by *Backofen and Will* (2001) to calculate optimal and suboptimal H-cores enables only the enumeration of connected cores. As discussed by *Will* (2005), it is possible to extend the CPSP approach such that optimal unconnected cores are implicitly generated via a set of optimal connected ones. But since we are only interested in modelling globular proteins, optimal structure will in almost all cases show a connected H-core (*Will*, 2005).

3.2.2 Extension to Side Chain Models

In the following, we will discuss the original CSP for backbone-only models formulated during the threading step of the CPSP approach. Based on this formulation, we will show how to extend the CPSP approach to side chain models to enable the prediction of optimal structures. The CSPs are given by $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, where we denote the set of variables $\mathcal{X}$, their domains $\mathcal{D}$, and a set of constraints $\mathcal{C}$ as introduced in Sec. 1.4.3.

CSPs for Backbone-only Models

The CSP formulated for backbone-only models within the second step of the CPSP approach can be defined as follows given a sequence S of length n, a lattice L, and an optimal H-core $h \in \mathcal{H}$.

For each sequence position S_i in S a variable $X_i \in \mathcal{X}$ is introduced representing the optimal structure $P = (P_1, \ldots, P_n)$ within a given lattice L. There is only one constraint to be obeyed: P has to be a valid structure, i.e. a self-avoiding walk. Within the CSP this is modeled by a sequence of binary neighboring constraints $\text{neigh}(X_i, X_{i+1})$ enforcing $(X_i - X_{i+1}) \in N_L$ and a global alldifferent($\mathcal{X}$) constraint (*Régin*, 1994) to enforce the self-avoidingness of P, i.e. $\forall i \neq j : X_i \neq X_j$. The optimal H-core $h \in \mathcal{H}$ is used to define the domains $D_i \in \mathcal{D}$: $\forall_{i:S_i=\mathtt{H}} : D_i = h$ and $\forall_{i:S_i=\mathtt{P}} : D_i = L \setminus h$. The application of the H-cores ensures that any solution of such a CSP, i.e. an assignment for each variable that satisfies all constraints in $\mathcal{C}$, will encode a structure that maximizes the number of HH-contacts and thus minimizes the HP energy function. Therefore, any solution results in an optimal structure of S.

The domains $D_i \in \mathcal{D}$ can be further constrained in order to reduce the search space of the CSP. To this end, we use the *hull level* function $\text{lvl} : 1 \leq i \leq n \to \mathbb{N}$ defined for each sequence position i from the given sequence S. The hull level is the minimal distance in sequence to a monomer of different type, i.e.

$$\text{lvl}(i) = \underset{k \geq 1}{\arg\min} \{S_{i-k} \neq S_i \vee S_{i+k} \neq S_i\} \,. \tag{3.1}$$

The hull level is computed for both `H`- and `P`-monomers. Thus, an `H`-monomer with hull level 1 has to be on the surface layer of the H-core, such that it can be neighbored to a `P`-monomer that cannot be placed within the core. On the other hand, a `P`-monomer with hull level 2 can be placed up to two lattice points outside of the core, i.e. it has to be placed within an H-core surrounding hull of "thickness" two. For a given H-core $h \in \mathcal{H}$, this enables a further constraining of the domains based on hulls recursively defined for each hull level by

$$\begin{aligned}\text{hull}^{\mathtt{H}}(l,h) &= \begin{cases}\{\ \vec{v} \mid \vec{v} \in h \wedge \exists_{\vec{n} \in N_L(\vec{v})} : \vec{n} \notin h\ \} & \text{if } l = 1\\ \text{hull}^{\mathtt{H}}(l-1,h) \cup \{\ \vec{v} \mid \vec{v} \in h \wedge \exists_{\vec{n} \in N_L(\vec{v})} : \vec{n} \in \text{hull}^{\mathtt{H}}(l-1,h))\ \} & \text{else, i.e. } l > 1\end{cases}\\ \text{hull}^{\mathtt{P}}(l,h) &= \begin{cases}\{\ \vec{v} \mid \vec{v} \in L \setminus h \wedge \exists_{\vec{n} \in N_L(\vec{v})} : \vec{n} \in h\ \} & \text{if } l = 1\\ \text{hull}^{\mathtt{P}}(l-1,h) \cup \{\ \vec{v} \mid \vec{v} \in L \setminus h \wedge \exists_{\vec{n} \in N_L(\vec{v})} : \vec{n} \in \text{hull}^{\mathtt{P}}(l-1,h)\ \} & \text{else, i.e. } l > 1\end{cases}\end{aligned}$$

Therefore, an enhanced domain initialization for the CSP from above would be

$$D_i = \begin{cases}\text{hull}^{\mathtt{H}}(\text{lvl}(i),h) & \text{if } S_i = \mathtt{H}\\ \text{hull}^{\mathtt{P}}(\text{lvl}(i),h) & \text{else, i.e. } S_i = \mathtt{P}\end{cases} .$$

Using this initialization, the variable domains are much smaller than in the initial CSP given above. This results in less pruning and lower runtimes. A further restriction of some domains can be achieved for positions where both adjacent monomers are of different type, so called *singletons*. Here, the initialization with $\text{hull}^{\mathtt{H/P}}(1,h)$ can be restricted to positions from $\text{hull}^{\mathtt{H/P}}(1,h)$ that have two neighbored positions in the H-core/surrounding.

Finally, we can use the hull level assignment to exclude whole H-cores from processing by counting the level occurrences within the sequence. If a sequence shows more indices with a certain hull level than the according hull contains positions, the H-core can be excluded. Thus, it has to hold

$$\forall_l : |\{\ i : \text{lvl}(i) = l\ \}| \leq |\,\text{hull}(l,h)|\ .$$

This check can be extended to singletons too. In consequence, many optimal H-cores can be ignored for a given sequence if they do not fulfill the needed hull sizes. By that, we do not have to construct and solve a CSP for each H-core as sketched above but only for appropriate H-cores. Since the number of H-cores can be high (which is later discussed in detail in Sec. 6.2), this exclusion and pruning reduces the overall computation time a lot.

CSPs for Side Chain Models

Originally, the CPSP approach was introduced by *Backofen and Will* (2006) for backbone-only models as described so far. Nevertheless, the constraint-based protocol enables its extension to side chain models, broadening its applicability.

There are two main differences of side chain models compared to backbone-only models: (a) within structures each amino acid is represented by two instead of one monomer (see Sec. 1.2.2.2) and (b) the contact-based energy function considers only interactions between the hydrophobic side chain monomers. The placements of all backbone monomers and of the polar side chains are ignored

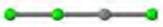
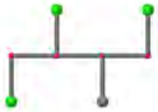
	Backbone-only Models 	Side Chain Models 	
Variables	one X_i for each S_i	X_i^b and X_i^s for each S_i	monomers
Domains	$D_i = \begin{cases} h & \text{if } S_i = \texttt{H} \\ L \setminus h & \text{if } S_i = \texttt{P} \end{cases}$	$D_i^b = L \setminus h$	backbone variables
		$D_i^s = \begin{cases} h & \text{if } S_i = \texttt{H} \\ L \setminus h & \text{if } S_i = \texttt{P} \end{cases}$	side chain variables
Constraints	alldifferent$(\mathcal{X})$	alldifferent$(\mathcal{X})$	self-avoidingness
	$\forall$: neigh$(X_i - X_{i+1})$	$\forall$: neigh$(X_i^b - X_{i+1}^b)$	backbone connectivity
		$\forall$: neigh$(X_i^b - X_i^s)$	side chain connectivity

Table 3.1: The differences of the CSP formulation for backbone-only and side chain models within the CPSP approach. Changes necessary to define the side chain models are highlighted in red. See text for further details.

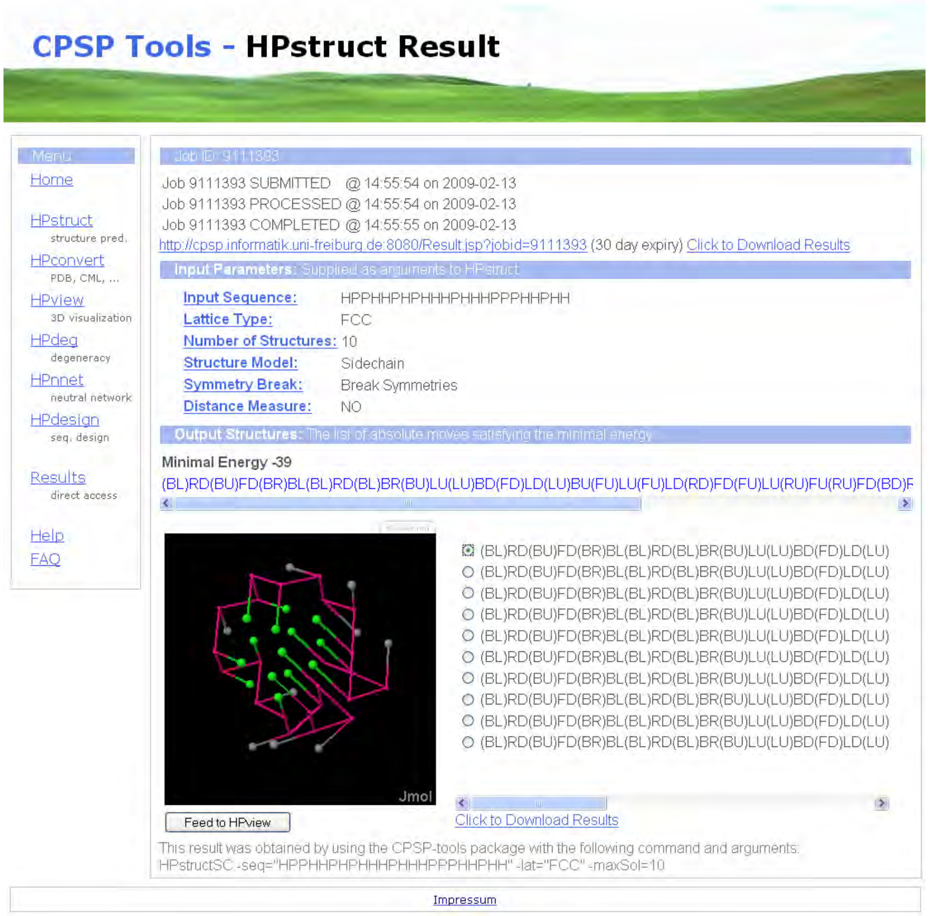
for the energy calculation (see Sec. 1.2.3).

A detailed analysis of the backbone-only CPSP approach reveals that a change of the CSP formulation suffices to adapt the approach to side chain models. The search procedure as well as the applied screening of appropriate optimal and suboptimal H-cores can be maintained without further changes.

Since the CSP from above defines one variable per monomer, we have to define two variables X_i^b and X_i^s for each sequence position S_i to represent its backbone P_i^b and side chain position P_i^s within the structure $P = ((P_1^b, P_1^s), \ldots, (P_n^b, P_n^s))$, respectively. In the backbone-only CSP, the energy of the resulting structures is implicitely encoded by the given H-core $h \in \mathcal{H}$ since all H-monomers are constrained to H-core positions. When considering the energy function for side chain models, we have to apply this constraint to hydrophobic side chains only as discussed above, i.e. $D_i^s = h$ if $S_i = \texttt{H}$, all other monomers have to be placed outside of the H-core $(L \setminus h)$. Among the constraints only the connectivity of backbone and side chain monomers has to be ensured additionally. Table 3.1 that opposes the two CSP definitions of both models. The necessary changes to yield CSPs for side chain models are highlighted in red.

Using the altered CSP for side chain models, the CPSP approach depicted in Sec. 3.2.1 and Fig. 3.2 can be applied without further changes (*Mann et al.*, 2009c).

When investigating optimal structures in the HP side chain model, one faces a restriction of our prediction approach: the H-cores formed by side chain monomers do not show fully enclosed positions since every hydrophobic side chain positioned in the core has to be connected to a (non-hydrophobic) backbone. Since the latter is constrained to be outside of the H-core, we can only predict optimal structures if we have cores without fully enclosed positions not reachable from the surface. We screened our database of H-cores in the 3D cubic lattice and found that up to an H-core



Job ID: 9111393

Job 9111393 SUBMITTED @ 14:55:54 on 2009-02-13
Job 9111393 PROCESSED @ 14:55:54 on 2009-02-13
Job 9111393 COMPLETED @ 14:55:55 on 2009-02-13
http://cpsp.informatik.uni-freiburg.de:8080/Result.jsp?jobid=9111393 (30 day expiry) Click to Download Results

Input Parameters: Supplied as arguments to HPstruct

Input Sequence:	HPPHHPHPHHHPHHHPPPHHPHH
Lattice Type:	FCC
Number of Structures:	10
Structure Model:	Sidechain
Symmetry Break:	Break Symmetries
Distance Measure:	NO

Output Structures: The list of absolute moves satisfying the minimal energy

Minimal Energy -39
(BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)BU(FU)LU(FU)LD(RD)FD(FU)LU(RU)FU(RU)FD(BD)F

● (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)
○ (BL)RD(BU)FD(BR)BL(BL)RD(BL)BR(BU)LU(LU)BD(FD)LD(LU)

Click to Download Results

This result was obtained by using the CPSP-tools package with the following command and arguments:

```
HPstructSC -seq="HPPHHPHPHHHPHHHPPPHHPHH" -lat="FCC" -maxSol=10
```

Figure 3.3: Screenshot of the result visualization of the CPSP-web-tools interface of HPSTRUCT (*Mann et al.*, 2008b), the CPSP approach implementation for optimal structure prediction. It is exemplified within the 3D FCC lattice using the side chain model extension presented in Sec. 3.2.2. Interactive structure visualization (lower left) for the selected resulting structure in absolute moves (lower right) is done via JMOL (*Herráez*, 2006).

size of $n_H < 25$, at least one optimal H-core shows no buried positions and is a candidate for a succesful threading. When considering suboptimal H-cores up to the second level of suboptimality, this bound increases to a core size of 35. Considering that a typical protein sequence is not completely hydrophobic, we should able to predict optimal side chain model structures in the 3D cubic lattice at least up to a protein length of 40. A similar problem occures if a suboptimal H-core shows an encapsulated empty position. This position is theoretically available for a backbone or P-monomer placement, but once again this would inhibit the backbone or side chain connectivity of the structure. But since such enclosed holes would significantly reduce the number of contacts within the H-core, we expect such configurations only for very suboptimal and large H-cores.

3.2.3 CPSP-tools and CPSP-web-tools

We have successfully implemented the CPSP approach and other HP model related tools presented within this thesis. All tools are bundled in the CPSP-tools package available at

`http://cpsp.informatik.uni-freiburg.de`

The CPSP-tools package provides a set of programs that enables typical, modern research tasks to be calculated efficiently and accurately. In the following, we list the programs each with a typical example application. HPstruct predicts (all) optimal and suboptimal structures as required for investigating properties of low energy conformations, as e.g. studied by *Jacob and Unger* (2007). The statistical analysis of protein-like sequences, see (*Blackburne and Hirst*, 2005), requires a degeneracy-based classification of sequences. This is possible with HPdeg that will be presented in detail within the next section. For the exploration of protein evolution, similar to (*Wroe et al.*, 2007), one needs to investigate the sequence-structure space. We provide HPdesign for sequence design and HPnnet for neutral network computation, both will be introduced in Chapter VI.

In addition to those described above, CPSP-tools provides a set of utility programs helpful for lattice protein studies. For instance using HPconv it is possible to convert between different structure encodings. Among them are the move strings introduced in Sec. 1.2.2.3 and 3D-position data in coordinate (XYZ-), Protein Data Bank (PDB-), and Chemical Markup Language (CML-) format. A move string normalization, as well as a conversion into an orientation independent relative move string, is available for a symmetry independent structure comparison (see Sec. 1.2.2.3). Finally, HPview interactively visualizes structures in different lattices using the Jmol interface (*Herráez*, 2006).

All methods can be applied to HP-sequences in the 3D cubic and the more complex face centered cubic lattice model to generate both, backbone-only and side chain models. The validity of the algorithms has been tested and confirmed on a large set of benchmark problems. The functionality of H-core database access, structure prediction, and degeneracy computation are collected in the CPSP C++ library. A complete application programming interface and source code documentation is included that enables the embedding, extension, and use of the CPSP approach in new programs and applications. The package supplies standard installation procedures for Linux based on common tools (GNU automake) and can be compiled and installed easily on current 32- and 64-bit Linux systems (including Cygwin^{TM} for Microsoft Windows^{TM}). All programs are written in C++ and use the state-of-the-art Constraint Programming System Gecode (Gecode) for highest performance. They provide a slim text-based user interface for efficient pipelining as required for high-throughput experiments.

Furthermore, we provide our CPSP-web-tools that enable direct instant access to the majority of the CPSP-tools when no high-throughput experiments are needed (*Mann et al.*, 2009c). We provide an extensive help area that covers frequently asked questions covering general information on lattice protein models as well as specific questions about the CPSP approach and tools. The web portal serves as a platform for research and teaching in the field of HP protein models. The combination of interactive result visualization including 3D views of the structures, the interlink of the different tools for chained applications and the fast runtimes of the CPSP-tools result in a useful service for the end-user. Currently, we encounter about 3-5 users a day on average (statistics by Google Analytics[3]). An example of the HPstruct web interface is presented in Fig. 3.3.

[3] Accessed at 4th May 2011 at `www.google.com/analytics/`

3.3 Optimal Structure Degeneracy

Publications

- **CPSP-web-tools: a server for 3D lattice protein studies**
  M. Mann*, C. Smith, M. Rabbath, M. Edwards, S. Will, and R. Backofen*
  in *Bioinformatics*, vol. 25 (5), pp. 676–677, 2009
  doi:10.1093/bioinformatics/btp034

- **Decomposition during search for propagation-based constraint solvers**
  M. Mann*, G. Tack, and S. Will*
  Technical report, ALU Freiburg, 2007
  arXiv:0712.2389

- **Counting protein structures by DFS with dynamic decomposition**
  S. Will and ***M. Mann***
  in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'06)*, pp. 83–90, 2006

If an energy function E of a protein model allows that two different structures yield the same energy value for a fixed sequence, the model is named *degenerated*. In the following, we define the *protein sequence S to be degenerated* if it exists more than one optimal structure with minimal energy $E_{\min}$. Formally we say

$$S \text{ is degenerated} \quad \longleftrightarrow \quad \exists_{P \neq \hat{P}} : E(S,P) = E(S,\hat{P}) = E_{\min} \quad \text{with } \forall_{\breve{P}} : E_{\min} \leq E(S,\breve{P}) \ . \tag{3.2}$$

The degeneracy $\deg(S)$ of a protein sequence S is defined as the number of minimal energy structures, i.e.

$$\deg(S) = |\{ \; P \mid E(S,P) = E_{\min} \; \}| \ . \tag{3.3}$$

The degeneracy of a protein is a measure of its thermodynamic stability (*Wiedersich et al.*, 2008; *Will*, 2005). A protein that shows a unique lowest energy structure and very few structures with nearly optimal energy is more likely to adopt the optimal structure within the thermodynamic equilibrium compared to a more degenerated protein. Thus, we can use the degeneracy as a rough guide to assess a protein's thermodynamics.

An important application of the degeneracy analysis is the classification of sequences as protein-like or not, an application discussed in detail in Sec. 5.5. A sequence S is protein-like if it can adopt only one optimal structure ($\deg(S) = 1$), a definition which is common in literature (*Crippen and Chhajer*, 2002; *Li et al.*, 2002; *Huard et al.*, 2006; *Jacob and Unger*, 2007). As a result, the functional fold of the protein is clearly defined as the unique minimal energy structure within the model and it will be formed with high stability according to stability.

In order to calculate the degeneracy of sequences the CPSP approach can be applied since it enables the enumeration of *all* optimal structure for a given sequence (see above). Therefore, a straight-forward scheme to calculate a protein's degeneracy is to count all solutions of the CSPs formulated within the CPSP protocol. For this purpose, we have implemented a dedicated program HPDEG that is part of the CPSP-package (*Mann et al.*, 2008b). Note, since the CPSP approach is based on precomputed *compact and connected* hydrophobic cores, the calculated degeneracy by HPDEG is a lower bound on the real degeneracy by missing some structures not showing a compact core. But since non-connected or looser H-monomer distributions usually directly result in less contacts, the bound is seen to be close to the full degeneracy in most cases (*Will*, 2005). Thus, we use the number of optimal structures with compact H-core computed by HPDEG as a (lower) estimate of $\deg(S)$.

But the degeneracy calculation can be even improved. We have shown in (*Will and Mann*, 2006) that the application of AND/OR-search can significantly speedup the approach. AND/OR-search was introduced by *Dechter and Mateescu* (2004) for Constraint Programming, generalizing the work of *Bayardo Jr. and Pehoushek* (2000) for 3-SAT. When solving a CSP with AND/OR-search the structure of the CSP is analyzed concerning the constraint dependencies between variables. If independent subproblems (not connected via any constraint) can be identified, they are handled separately and their partial solutions are later combined to whole solutions. For details of the method and how to apply to protein structure prediction please refer to (*Mann*, 2006; *Will and Mann*, 2006; *Mann et al.*, 2007a). Note, since the Gecode Constraint Programming library (Gecode) used by the CPSP-tools currently does not support AND/OR-search, our tool HPDEG currently only implements the standard search and solution counting.

In the following, we study the degeneracy distribution within the HP model's sequence space using HPDEG for both backbone-only and side chain models. To this end, we generated a large number of random HP-sequences of length 27 and determined their degeneracy in the 3D cubic lattice. In order to reduce computation time, we limited the degeneracy calculation to an upper bound of one million, i.e. for sequences with higher degeneracy only a lower bound of "$\deg(S) > 10^6$" is computed.

3.3.1 Degeneracy in Backbone-only Models

The degeneracy distribution for 100,000 randomly generated sequences for backbone-only models is visualized in Fig. 3.4. The majority of sequences can adopt more than 1,000 optimal structures where the median of the distribution is clearly larger than 10^4.

Only 16 out of the 100,000 sequences tested are *non-degenerated*, i.e. can adopt only a unique optimal structure ($\deg(S) = 1$). When ignoring the 4,358 sequences where no degeneracy could be calculated due to our restricted H-core database (see Sec. 3.2.1), we can estimate that only about 0.01-0.02% of the sequence space $\left(\frac{16}{96{,}642} = 0.0165\%\right)$ is non-degenerated. The fraction of about 0.02% is consistent with further studies for other sequence lengths in the 3D cubic lattice not shown in this manuscript. When comparing the 3D results with the 2D-square lattice study by *Bornberg-Bauer* (2002), we can see the increase in structural flexibility within the 3D-lattice. Within his study, *Bornberg-Bauer* (1997b, 2002) has used an exhaustive enumeration of sequence and structure space to identify non-degenerated sequences of length 18 in the unrestricted 2D-square lattice. He found 2.4% of the sequences to be non-degenerated reproducing the results of *Chan and Dill* (1996). This percentage was shown to be stable for different sequence lengths (see the review by *Chan and Bornberg-Bauer* (2002)) and is more than 100-times the number than for the 3D-cubic lattices as

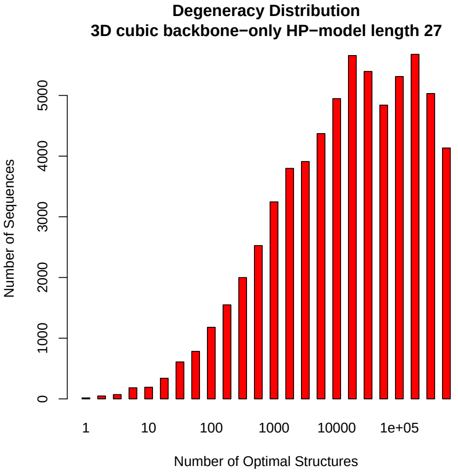


Figure 3.4: Degeneracy distribution of backbone-only HP models exemplified in the 3D cubic lattice for sequence length 27. The distribution is only shown for degeneracy below 10^6, see text for further information.

we have shown.

The number of sequences of length 27 with a *degeneracy above one million* is high (29,828) and comprises about 30% of sequence space $\left(\frac{29{,}828}{96{,}642} = 30.86\%\right)$. Usually, such sequences show a low number of H-monomers and long stretches of Ps within their sequence. By that, large parts of these structures are unconstrained by the energy function resulting in a high flexibility among optimal structures.

Since the number of non- or low-degenerated sequences is very low, a randomized identification is almost impossible. *Will* (2005) has shown, that a local search strategy can be applied in order to search sequence space for low degenerated sequences. In short, the algorithm follows a Markov Chain Metropolis scheme (see Sec. 1.4.2.1) utilizing the logarithm of the degeneracy estimates as pseudo-energies. He could show that a fast identification of low degeneracy structures is possible. The approach was implemented in HPoptdeg and is part of the CPSP-tools package (*Mann et al.*, 2008b).

3.3.2 Degeneracy in Side Chain Models

Since the CPSP approach for side chain models is computationally more demanding than for backbone-only models, we studied the degeneracy distribution of side chain models for a reduced number of 10,000 random sequences.

Only 408 sequences show a *degeneracy below one million*. When ignoring the 311 sequences where the degeneracy could not be calculated, this comprises only about 4% of the tested sequences $\left(\frac{408}{9{,}689} = 4.21\%\right)$! Comparing to the 70% for backbone-only models, this shows the immense increase in structural flexibility and thus degeneracy for side chain models. As for high-degenerated backbone-only models, the vast numbers arise from the low confinement of the structure by the energy function. Only the hydrophobic side chains are constrained by the energy model. More than half of the structure (the whole backbone as well as P side chain monomers) are only constrained by the structural connectivity and self-avoidingness, leaving much room for structural diversity without changing the energy.

Among the 10,000 sequences tested, the lowest degenerated sequence found still enabled 264 optimal structures. Using the HPoptdeg implementation for side chain models by *Rabbath* (2008), the sequence with the lowest degeneracy encountered could fold still 48 non-symmetric optimal structures. So far, we have not found any non-degenerated HP sequence of length 27 in the 3D cubic side chain lattice model.

3.4 Optimal Structure Grouping

> Publication
>
> **Equivalence Classes of Optimal Structures in HP Protein Models Including Side Chains**
> ***M. Mann****, R. Backofen and S. Will*
> in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'09)*, 2009
> arXiv:0910.3848

As we have shown above, the hydrophobicity-focusing energy function in HP models results on average in a vast number of optimal structures. Since polar residues do not contribute to the energy, optimal structures usually show a much higher variation in the placement of polar than hydrophobic residues. This is even more severe in side chain models where the majority of the monomers is unconstrained by the energy function, since both all backbone monomers as well as polar side chains are not considered.

Here, we introduce an equivalence relation to partition the set of (optimal) structures into according classes. Two structures are defined to be equivalent, iff they do not differ in the relative placements of their hydrophobic residues. This is exemplified in Fig. 3.5. We introduce an extension to the CPSP approach that enables an efficient calculation of the number of equivalence classes of optimal structures via the enumeration of one representative per class. The approach is presented for backbone-only and side chain HP models. We show that a sequence's number of representatives (later defined as core-degeneracy) is several magnitudes smaller than the overall number of all optimal structures (degeneracy).

Thus, the set of optimal representatives is well placed to be used when investigating more detailed protein models, as done within the combined approach of CPSP and local search by *Ullah et al.* (2009) described in Sec. 3.5. Furthermore, we propose another application of the equivalence classes: Since the equivalence relation is highly correlated to the HP energy function, the number of classes might be a better measure of structural stability than a sequences' degeneracy, e.g. used by *Shortle et al.* (1992).

3.4.1 Method

Revisiting the CSP for the CPSP approach from Sec. 3.2.1 we can see that P monomers are constrained only by fself-avoiding walk (SAW) constraints. Imagine a sequence with a long tail of P monomers. Each valid placement of the subchain in front of the tail can be combined with a combinatorial number of possible SAWs of the tail. This leads to the immense degeneracy in the HP model shown in Sec. 3.3.

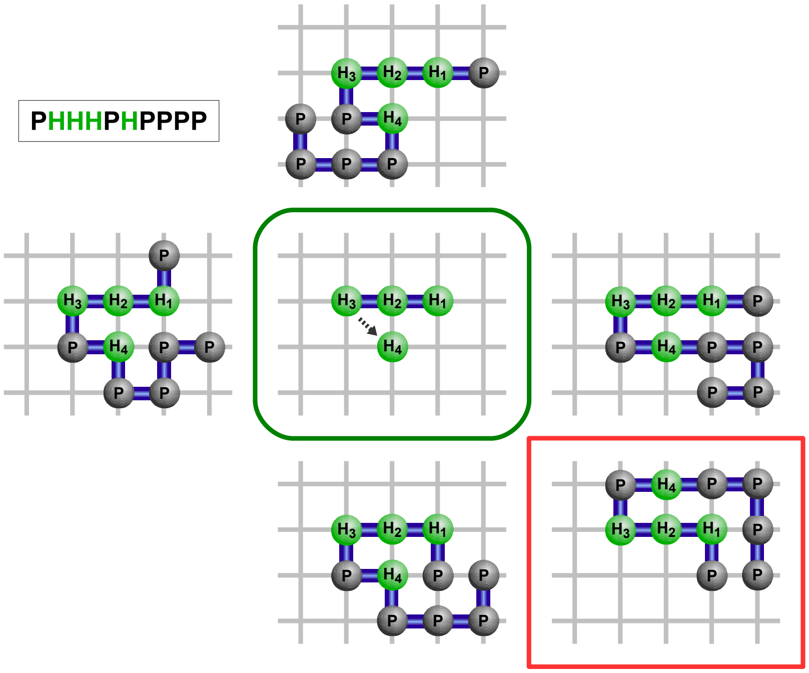


Figure 3.5: Different optimal structures for the sequence `PHHHPHPPPP` that show the same relative H-monomer placement (center), i.e. are within one equivalence class. The structure in the lower right highlighted in red is part of the equivalence class as well when considering its reflection on the X-axis. This exemplifies the symmetry problem for equivalence detection.

Therefore, we set up an *equivalence relation* $\overset{H}{\sim}$ on structures (Eq. 3.4) that decomposes the set of all (optimal) structures into equivalence classes. In the following, the number of equivalence classes of optimal structures is denoted as *core-degeneracy*. As given by Eq. 3.4, structures from different equivalence classes differ in at least one H monomer placement. Formally, two backbone-only structures $P = (P_1, \ldots, P_n)$ and $\hat{P} = (\hat{P}_1, \ldots, \hat{P}_n)$ are said to be equivalent if and only if

$$P \overset{H}{\sim} \hat{P} \longleftrightarrow \exists_{r\in\mathcal{R}}\ \exists_{t\in L} : \forall_{i|S_i=\mathtt{H}} : P_i = \hat{P}_i R_r + t\ , \tag{3.4}$$

where $\mathcal{R}$ denotes the set of all symmetry functions (according to rotation and reflection) within the underlying lattice L while R_r represents the according rotation/reflection matrix for the symmetry $r \in \mathcal{R}$ (see Sec.1.2.2.3 for details). The translation vector $t \in L$ is needed to shift the symmetric structure back.

The representative enumeration (that corresponds to core-degeneracy calculation) can be done via an extension of the CPSP approach presented in Sec. 3.2.1. Instead of calculating all optimal structures, we want to calculate only one representative per equivalence class. This has to be ensured at two stages:

(I) the solutions of each CSP for a given H-core have to be different according to Eq. 3.4, and

(II) the solutions from two CSPs for two different H-cores have to be different as well.

The second condition (II) holds by definition, because $\overset{H}{\sim}$ is only defined on the H monomer placements that are constrained by different H-cores from the set of all optimal H-cores $\mathcal{H}$. Thus, they differ in at least one placement. In the following, we will discuss how to achieve the difference for solutions of a single CSP (I) that handles the threading through a single H-core.

Note that the core-degeneracy, i.e. the number of different placements of H-monomers, or core-configurations, in optimal structures of a sequence, is *not equal* to the number of different H-cores ($|\mathcal{H}|$), which are the sets of lattice points that are occupied by H-monomers. The latter number is easily obtained from the standard prediction algorithm, described in Sec. 3.2.1. It equals the number of cores, where the sequence is successfully threaded on, since it is possible to get several different threadings for a single H-core. The core degeneracy is usually much larger than the number of used cores.

Restricted Search for Enumeration of Representatives

The standard way to solve a CSP is a combination of domain filtering (i.e. constraint propagation) and depth-first search. This results in a binary search tree where each node represents a subproblem of the initial CSP (root) and edges show the additional constraints added to derive the two subproblems from its predecessor node/CSP. The constraints c and $\neg c$, added to derive the leave nodes of a certain CSP, are often of the form $c = (X_i \equiv d)$ by selecting a variable X_i from $\mathcal{X}$ and a value $d \in D_i$ according to some heuristics (see Sec. 1.4.3). The constraint solver traverse the binary tree until a solution was found or a constraint from $\mathcal{C}$ was violated.

Therefore, a straightforward way to enumerate only one representative for each equivalence class can be sketched as follows: first, we restrict the search of the solving process onto the H associated variables. Then, we perform a single check for satisfiability, i.e. search for a single assignment of P monomer variables fulfilling all constraints in $\mathcal{C}$. Thus, we get only one P monomer placement for a given H monomer assignment if any exists.

The drawback of this approach is that we restrict the variable order of the search heuristics. But the performance of the CPSP approach mainly depends on the search heuristics applied to select a certain variable or value from its domain. It turned out that a mixed assignment of H and P associated variables yields the best runtimes. These heuristics can not be applied within the sketched procedure where we have to first assign H-associated variables, then P-associated ones. Thus, a lower CPSP performance is expected. Nevertheless, the sketched approach results in less search effort (and thus runtime) than an enumerating all optimal structures followed by a brute-force classification.

We have implemented the approach resulting in the tool HPrep, now part of the CPSP-tools version 2.4.0. It implements both the enumeration of equivalence classes with one representative structure per class as well as the pure calculation (counting) of the core-degeneracy. Furthermore, backbone-only as well as side chain models are supported where the approach for the latter is discussed next.

Representative Optimal Structures with Side Chains

As shown in Sec. 3.2.2, the CPSP approach can be extended to side chain models (*Mann et al.*, 2009c). Here, each amino acid of a protein sequence is represented by two monomers: P_i^b representing the backbone atoms, and P_i^s representing the atoms of the side chain. Beneath the SAW condition on the backbone monomers P_i^b, we constrain each side chain to be neighbored to its backbone, i.e.

$\forall_{1\leq i\leq n} : (P_i^s - P_i^b) \in N_L$. The applied energy function exploits only HH-contacts of side chain monomers P_i^s (see Sec. 1.2.3.1)

We have shown in the previous Sec. 3.3 that side chain models show an even higher degeneracy than the backbone-only models, since all backbone monomers P_i^b as well as polar side chain monomers are unconstrained by the energy function. Thus, an equivalence relation $\overset{H}{\approx}$ that focuses on the monomers constrained by the energy function is even more striking in HP models including side chains. The relation $\overset{H}{\approx}$ for two side chain structures $P = ((P_1^s, P_1^b), \ldots, (P_n^s, P_n^b))$ and $\hat{P} = ((\hat{P}_1^s, \hat{P}_1^b), \ldots, (\hat{P}_n^s, \hat{P}_n^b))$ is given by

$$P \overset{H}{\approx} \hat{P} \longleftrightarrow \exists_{r\in\mathcal{R}}\ \exists_{t\in L} : \forall_{i:S_i=\mathtt{H}} : P_i^s = \hat{P}_i^s R_r + t \tag{3.5}$$

in accordance with Eq. 3.4, i.e. R_r is the rotation/reflection matrix for a symmetry $r \in \mathcal{R}$ from the set of all symmetry functions (according to rotation and reflection) within lattice L and $t \in L$ is a translation vector.

Therefore, we will enforce that structures from one equivalence class show the same relative H monomer side chain positioning. The CPSP approach for HP models including side chains differs only in the CSP formulation from the original approach for backbone-only models (see Sec. 3.2.2). This allows for the application of the same approach discussed in the previous section to enumerate non-equivalent optimal structures that are representatives for each equivalence class. Thus, we restrict search to the H associated side chain variables first and only check for satisfiability on the remaining variables.

In the following, we will study the distribution of the resulting core-degeneracy.

3.4.2 Results and Discussion

We exemplify the enumeration of representatives for backbone-only and side chain models. We focus on the comparison of the resulting core-degeneracy of a sequence and its overall number of optimal structures, i.e. degeneracy, because we are interested in a reduced set of optimal structures, e.g. for local search initialization (see (*Ullah et al.*, 2009)).

All following results are given for HP-sequences of length 27 in 3D cubic lattice. Since the enumeration and check of all 2^{27} sequences ($> 10^8$) is computationally not feasible, we restrict each study to the large randomly chosen subset of 10^5 and 10^4 sequences that were used in the degeneracy study from the previous Sec. 3.3. This enables a direct comparison of the newly defined core-degeneracy distributions with the classic degeneracy distributions discussed above. Thus, for each sequence from the according set from the degeneracy study we run HPrep to calculate the core-degeneracy of the sequence. The results are presented and discussed in the following.

3.4.2.1 Representatives in Backbone-only Models

We tested the 10^5 random sequences from the backbone-only degeneracy study from Sec. 3.3.1 in the 3D cubic lattice. Here, only 66% show a degeneracy below 10^6. For about 4% of the sequences the current H-core database was not sufficient to determine optimal structures (see CPSP approach Sec. 3.2.1 for further details). The remaining 30% can adopt even more than 10^6 structures with minimal energy.

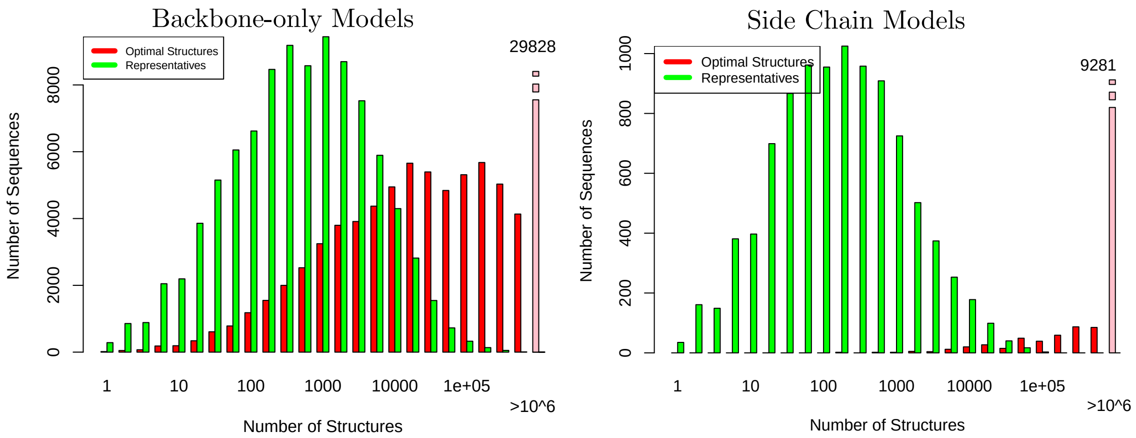


Figure 3.6: Distribution of the number of representatives (core-degeneracy) versus the overall number of optimal structures (degeneracy). They are exemplified for for HP models in the 3D cubic lattice for sequence length 27. On the [left], results correspond to backbone-only predictions for 100,000 random sequences, the [right] figure depicts side chain model predictions for 10,000 random sequences. The distributions are only shown with exact values for degeneracy below 10^6, all sequences with a higher degeneracy are collected in the right most bar in pink representing "$> 10^6$".

Figure 3.6 (top) summarizes the results of the representative study in backbone-only models. The figure presents in *red* the degeneracy distribution (see Sec. 3.3.1) and in *green* the core-degeneracy distribution (i.e. number of representatives) both with cut-off 10^6. Sequences with a degeneracy or core degeneragy above the cutoff a represented by a single bar on the right with the according value. Thus, in *red* the degeneracy distribution with values below the cut-off comprises only 66% of the tested sequences as given above. In contrast, *all* sequences show a number of optimal equivalence classes below 10^6 (in *green*)! The average degeneracy is reduced from $124,800$ (with cutoff 10^6) to a mean core-degeneracy of $4,856$. This reduction within two orders of magnitude results in reasonably small sets of representative structures e.g. to be utilized in local search initializations that we will discuss later in Sec. 3.5. Furthermore, the enumeration of representatives is on average six times faster than the enumeration of all optimal structures with a mean runtime of 2 seconds (Opteron 2,356 - 2.3 GHz). Slightly higher runtimes can occur due to the changed search heuristic (see Sec. 3.4.1). A visualization of the runtime comparison is given in Fig. 3.7.

The large number of small sets of representatives compared to the usually immense complete sets of optimal structures shows the advantage of the approach: core-degeneracy does not show the huge combinatorial explosion of degeneracy. This gets even more striking in HP models including side chains, as shown in the next section.

3.4.2.2 Representatives in Side Chain Models

The degeneracy in HP models including side chains is much higher than for backbone-only models. This results from the simple energy function that does not constrain the backbone or P-monomers, see Eq. 1.20 in Sec. 1.2.3. Therefore, an immense number of optimal structures is present. From the 10^4 random HP-sequences tested only 408 show a degeneracy below 10^6. Again, for about 3.1% of the sequences no optimal structure was computed due to a restricted H-core database.

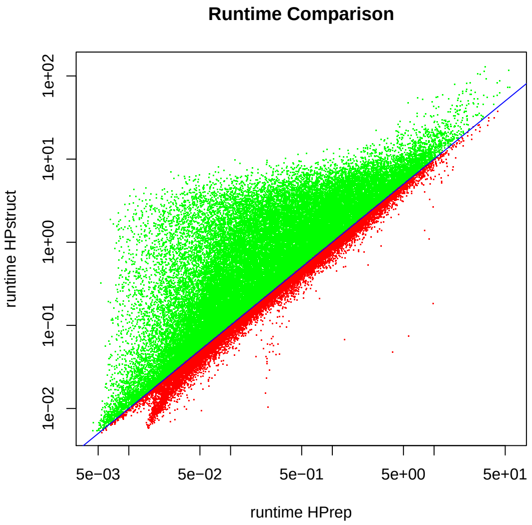


Figure 3.7: Runtime comparison of HPSTRUCT (Y-axis) and HPREP (X-axis) given in seconds and logarithmic scale. Dots in green mark sequences where the core-degeneracy calculation via HPREP was faster, red colored dots mark faster runtimes of HPSTRUCT. The numbers refer to the backbone-only HP model data in the 3D cubic lattice of sequence length 27 (see text).

When investigating core-degeneracy the picture changes completely: *All* of the sequences tested have less than 10^6 representatives. Figure 3.6 (bottom) summarizes the distribution. The average number of representatives is about 1,550, which is again at least three orders of magnitude smaller than the average degeneracy. Since we have only a very rough lower bound of 10^6 on the average degeneracy (due to the cut-off), the real reduction ratio is expected to be even higher.

3.4.3 Conclusions

The introduced equivalence relations for HP models enable an energy function driven partitioning of the structure space. The presented CPSP approach extension enables an efficient calculation of representatives for each equivalence class of optimal structures, i.e. the calculation of a sequence's core-degeneracy. Using our implementation HPREP, we showed that sequences have several orders of magnitude less optimal equivalence classes than optimal structures. This is most striking in models including side chains.

The sets of representatives are usually small. Furthermore, representatives show different hydrophobic core arrangements. Therefore, they are well placed to be used for the initialization of local search procedures that utilize more complex energy functions (*Ullah et al.*, 2009). The approach is discussed in more detail in the following section 3.5.

Since a sequence's degeneracy is a measure of structural stability (*Shortle et al.*, 1992), we propose another application of our approach. The core-degeneracy might be used as a more reasonable *measure of stability* in the HP model compared to degeneracy. It ignores the HP model specific degeneracy blow-up due to unconstrained subchains of P monomers (see section 3.4.1). Thus, a structural stability analysis could be based on the presented equivalence classes instead of all possible structures.

Finally, we can use the equivalence relation to define a *fold* class of structures of different sequences to be used within evolutionary studies. An according approach is discussed in detail in Sec. 6.4.

3.5 CPSP-guided Local Search

Publication

Protein Folding Simulation by Two-stage Optimization
A. D. Ullah, L. Kapsokalivas, ***M. Mann*** *and K. Steinhöfel*
in *Proc. of the International Symposium on Intelligence Computation and Applications (ISICA'09), CCIS*, vol. 51, pp. 138-145, Springer, 2009
doi:10.1007/978-3-642-04962-0_16

In nature, proteins fold very rapidly despite the enormous number of possible configurations. This observation is known as the Levinthal paradox and implies that protein folding can not be a random search for the global energy minimal structure (*Levinthal*, 1968).

One of the driving forces in folding of water-soluble globular proteins are hydrophobic interactions resulting in a compact packing of hydrophobic amino acids in the center of the protein. This effect can be captured by the HP model, introduced in Sec. 1.2.3.1, by considering only whether an amino acid is hydrophobic or polar (*Dill et al.*, 1995). Therefore, compact optimal structures in the HP model can be considered to represent a protein shortly after the hydrophobic collapse. When the preliminary hydrophobic core is formed, the protein folds into it's final functional/native fold. Such folding is typically modelled using local search schemes applying advanced energy models. Thus, we suggest and test a two-stage approach to mimic protein folding: (step 1) The hydrophobic collapse resulting in compact structures with hydrophobic cores is represented by the calculation of HP-optimal structures. (step 2) The compact structures from step 1 are used to initialize a local search scheme in order to identify the energy minimal functional fold. We compare the performance of this biologically motivated folding scheme with standard local search procedures.

Within this chapter, the CPSP approach by *Backofen and Will* (2006) has been widely discussed. It enables a very fast calculation of energy minimal structures within the HP-energy model needed for step 1. To be efficient, the CPSP approach it is restricted to HP-like energy models, e.g. the HPNX model (*Backofen et al.*, 1999; *Will*, 2005). For more sophisticated energy models, as the 20 × 20 pairwise potentials by *Miyazawa and Jernigan* (1985, 1996), no efficient constraint programming approach is known so far. Here, usually local search strategies are applied.

Local search is a generic optimization scheme that minimizes a given objective function. Given a neighborhood relation within the search space to traverse, local search is able to follow the gradient to identify local or even global minima (*Kirkpatrick et al.*, 1983; *Černý*, 1985). Marko Chain Monte Carlo local search schemes and Simulated Annealing are introduced in Sec. 1.4.2. Previous approaches using local search methods for protein structure prediction include tabu search (*Lesh et al.*, 2003; *Böckenhauer et al.*, 2008), genetic algorithms (*Berenboym and Avigal*, 2008), simulated annealing (*Li*, 2007; *Albrecht et al.*, 2008), ant colonization algorithms (*Song et al.*, 2006), and population-based local search methods (*Kapsokalivas et al.*, 2009).

Local search approaches work well in practice for more sophisticated energy functions but usually require a large number of steps to converge. Each search usually starts with a random start conformation to enable a broad search within structure space. In this paper we aim to combine the advantages of both approaches, i.e. the CPSP-method and local search procedures.

We introduce a protein folding simulation procedure that employs two stages of optimization in order to find structures of minimal energy. The method mimics the hydrophobic collapse during protein folding, i.e. the hydrophobicity-driven, fast formation of an initial structure and the following folding into the functional fold. Thus, the input protein sequence is first collapsed into a compact HP-optimal structure using the CPSP approach. Successively, the CPSP output is given as input to a Simulated Annealing-based local search procedure which employs the pairwise 20×20 energy potentials introduced by *Berrera et al.* (2003). The study is exemplified for backbone-only models and uses the 3D-FCC lattice that was shown to yield very good approximations of real protein structures, see Chapter II.

In the following, we will introduce the procedure details. A comparison to a standard Simulated Annealing scheme initialized with random start conformations will be used to evaluate the protocol.

3.5.1 Two-stage Optimization Approach

The approach works in two phases, namely the hydrophobicity optimization using constraint programming and the local search for the final energy minimization.

3.5.1.1 Hydrophobicity Optimization

In order to emulate the hydrophobic collapse, the given protein sequence S_{orig} has to be converted into a sequence S_{HP} within the HP model. For that purpose, we apply the hydrophobicity assignment table from Sec. 1.2.3.1 (Tab. 1.3) to convert each sequence position from the 20 letter amino acid alphabet into `H` or `P`.

The Constraint-based Protein Structure Prediction (CPSP) approach enables the calculation of optimal structures for these sequences in the HP-model. Using its implementation HPSTRUCT from the CPSP-tools package (*Mann et al.*, 2008b), we enumerate for a given sequence S_{HP} a representing set of optimal structures, all showing a compact hydrophobic core and shape.

The CPSP approach is discussed in detail in Sec. 3.2.1 and enables the enumeration of *all optimal structures*. These structures are used in the following phase to initialize the local search procedure.

3.5.1.2 Local Search

Simulated Annealing was introduced as an optimization tool independently by *Kirkpatrick et al.* (1983) and *Černý* (1985) and is presented in Sec. 1.4.2.2. To solve the PSP problem, we want to traverse the structure space of the protein, i.e. the set of all possible lattice protein structures. Thus, we have to define the neighboring system $\mathcal{N}$ (see Sec. 1.4.2.1) within the structure space that allows for a search space traversal with Simulated Annealing. Here, we apply the pull-move neighborhood for triangular lattices (*Böckenhauer et al.*, 2008), originally introduced by *Lesh et al.* (2003) for rectangular lattices.

In particular, Simulated Annealing requires an appropriate setting of the temperature cooling scheme in order to control the transitions to worse energies. In this study, we apply a logarithmic

PDB ID:	4RXN	S_{orig}:	MKKYTCTVCGYIYDPEDGDPDDGVNPGTDFKDIPDDWVCPLCGVGKDEFEEVEE
Length:	54	S_{HP}:	HPPHPHPHHPHHHPPPPPPPPPPPHPPPPPHPPHPPPHHHPHHPHPPPPHPPHPP
PDB ID:	1ENH	S_{orig}:	RPRTAFSSEQLARLKREFNENRYLTERRRQQLSSELGLNEAQIKIWFQNKRAKI
Length:	54	S_{HP}:	PPPPPHPPPPHPPHPPPHPPPPHHPPPPPPPPHPPPHPHPPPPHPHHHPPPPPPH
PDB ID:	4PTI	S_{orig}:	RPDFCLEPPYTGPCKARIIRYFYNAKAGLCQTFVYGGCRAKRNNFKSAEDCMRTCGGA
Length:	58	S_{HP}:	PPPHHHPPPHPPPHPPPHHPHHHPPPPPHHPPHHHPPHPPPPPPPHPPPPPHHPPHPPP
PDB ID:	2IGD	S_{orig}:	MTPAVTTYKLVINGKTLKGETTTKAVDAETAEKAFKQYANDNGVDGVWTYDDATKTFTVTE
Length:	61	S_{HP}:	HPPPHPPHPHHHPPPPHPPPPPPPPPHPPPPPPPPPHPPHPPPPPHPPHHPHPPPPPPPHPHPP
PDB ID:	1YPA	S_{orig}:	MKTEWPELVGKAVAAAKKVILQDKPEAQIIVLPVGTIVTMEYRIDRVRLFVDKLDNIAQVPRVG
Length:	64	S_{HP}:	HPPPHPPHHPPPHPPPPPHHHPPPPPPPHHHHPHPPHHPHPHPHPPHPHHHPPHPPHPPHPPHP
PDB ID:	1R69	S_{orig}:	SISSRVKSKRIQLGLNQAELAQKVGTTQQSIEQLENGKTKRPRFLPELASALGVSVDWLLNGTSDSNVR
Length:	69	S_{HP}:	PHPPPHPPPPHPHPHPPPPHPPPHPPPPPPPHPPHPPPPPPPPPPHHPPHPPPHPHPHPHHHPPPPPPPHP
PDB ID:	1CTF	S_{orig}:	AAEEKTEFDVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAPAALKEGVSKDDAEALKKALEEAGAEVEVK
Length:	74	S_{HP}:	PPPPPPPHPHHHPPPPPPPPHPHHPPHPPPPPPHPHPPPPPHHPPPPPPPHPPPHPPPPPPPPHPPPHPPPPPPPHPHP

Table 3.2: Benchmark sequences from Protein Data Base (PDB) and the derived HP-sequences.

cooling schedule that was shown to converge to optimal solutions (*Hajek*, 1988).

Within a standard Simulated Annealing protocol each search is started at an arbitrary point within the search space, i.e. a random structure from the structure space. Since the structure space growth exponentially, see Fig. 1.8 and Tab. 1.2, a high number of search steps is necessary in order to find close to optimal solutions.

Here, we want to incorporate more biological knowledge into the search. To this end, we initialize the local search with HP-optimal structures from the first phase that show already a compact hydrophobic core as it would be present after a hydrophobic collapse of the protein. Since their number can be very large still (see Sec. 3.3.1), we do a random uniform selection of start structures from this set of HP-optimal structures. The objective function to be optimized is the empirical contact potential described by *Berrera et al.* (2003), which is a 20 amino acid pairwise interactions energy function.

3.5.2 Results

In the following, we compare the performance of the two-stage optimization presented above to a standard optimization procedure based on randomly initialized local search alone. Although the new approach involves the CPSP approach in addition to local search, in practice the CPSP tool's runtime is very short and can be neglected (*Mann et al.*, 2008b). Thus, the performance of each method depends on the minimal energy reached, given a limited number of local search steps.

In other words, we examine the performance of Simulated Annealing starting from HP-optimal structures versus its performance with a randomly generated set of initial structures.

Table 3.2 shows the benchmark sequences used for the experiments, their PDB IDs, and their corresponding derived HP sequences. Benchmarks 4RXN, 4PTI, 1R69, and 1CTF are taken from (*Berrera et al.*, 2003). *Berrera et al.* (2003) have shown that their new empirical contact potential (employed in our approach) is able to discriminate the native structures of these 4 benchmark sequences. Benchmarks 1ENH, 2IGD, and 1YPA are taken from (*Dal Palu et al.*, 2004).

For each protein sequence we performed 10 independent local search iterations starting with random initial structures. Then we performed 10 independent runs for the two-stage approach

PDB ID	Method	Avg. E(Start)	Avg. E(End)	Min. E	Avg. Steps
4RXN	SA-only	-2.405	-161.625	-165.401	1,019,588
	2-stage	-140.377	-164.483	-167.781	816,844
1ENH	SA-only	-2.395	-149.456	-152.747	926,785
	2-stage	-127.347	-151.360	-153.098	904,368
4PTI	SA-only	-3.480	-208.969	-215.698	1,056,287
	2-stage	-179.196	-210.357	-212.500	652,600
2IGD	SA-only	-2.561	-178.941	-180.893	1,160,557
	2-stage	-163.201	-182.564	-183.205	706,773
1YPA	SA-only	-3.145	-252.556	-256.017	1,004,750
	2-stage	-236.895	-256.504	-257.810	1,142,827
1R69	SA-only	-3.055	-202.338	-215.166	1,073,051
	2-stage	-188.966	-216.708	-219.402	1,001,264
1CTF	SA-only	-1.804	-221.713	-228.921	1,176,490
	2-stage	-176.088	-231.225	-233.860	1,043,517

Table 3.3: Comparison between a standard Simulated Annealing scheme (SA-only) and our new two-stage optimization scheme (2-stage). Compared are the average energy of the initial structure ("Avg. E(Start)"), the average energy of the final structure ("Avg. E(End)"), the lowest energy seen among all iterations ("Min. E"), and the average number of steps performed per local search ("Avg. Steps").

where the initial HP-optimal structures are calculated with the CPSP-tool HPstruct.

The initial temperature for simulated annealing in both approaches was set equal to $D * \ln(2)$, where D is an estimation for the maximum depth of local minima of the underlying energy landscape. In a similar fashion to (*Albrecht et al.*, 2008), D was set equal to $n^{2/3}/c$, where n is the sequence length and c was chosen to be 1.5. Moreover, the maximum number of steps of each run in local search phase was limited to 1,500,000 for both approaches.

In Table 3.3 we present the results for each protein sequence. The first row (SA-only) corresponds to the results observed from local search alone starting from random structures, whereas the second row (2-stage) corresponds to the results observed from the two-stage optimization scheme starting from HP-optimal structures.

As we can see from column "Avg. E(Start)" in Table 3.3, the average energy of HP-optimal structures using the empirical contact potential is up to two orders of magnitude lower than the average energy of random structures. Given the same maximal number of steps to both approaches, the two-stage optimization always leads to conformations of lower energy on average compared to Simulated Annealing alone ("Avg. E(End)"). Also, the two-stage optimization reached lower best energy conformations ("Min. E") within the time limit for all benchmarks except 4PTI. Moreover, it requires on average less number of steps to produce conformations within the average final energy level ("Avg. Steps"), except for benchmark 1YPA. The shorter search might result from the already close energy of the initial HP-optimal start conformations compared to the final energies (compare column "Avg. E(Start)" with "Min. E").

In general, the two-stage optimization approach outperforms standard Simulated Annealing, since it reaches better final conformations in fewer iterations for the majority of benchmarks.

3.5.3 Conclusions

The introduced two-stage optimization approach for protein structure prediction combines the advantages of Constraint-based Protein Structure Prediction (CPSP) and local search. CPSP is very efficient for the HP model but computationally infeasible for a 20 amino acid pairwise interaction energy function. At the same time, local search methods are applicable to the latter problem, despite the considerable amount of computational effort required. Linking both approaches enables a combination of their advantages, i.e. an increased effciency of local search via a constraint-based search initialization.

Experimental results with real protein sequences show that the CPSP-tool HPstruct produces compact structures with an energy for the pairwise energy function that is up to two orders of magnitude better than the energy of a randomly generated structure. These compact structures that maximize hydrophobic interactions are used to initialize a Simulated Annealing-based local search procedure. The experiments show that better (lower energy) structures are obtained in fewer iterations compared to a standard Simulated Annealing protocol with a random initialization.

The proposed two-stage optimization outperforms a standard local search procedure based on simulated annealing alone. This highlights the importance of the incorporation of biological knowledge into protein structure prediction approaches since our two-stage scheme mimics the hydrophobic collapse observed in real protein folding.

3.6 Discussion and Future Work

The Protein Structure Prediction (PSP) problem is a hard computational problem in Bioinformatics. The Constraint-based PSP (CPSP) approach by Backofen and Will is currently the only method to solve the problem exact and efficiently.

Using the state-of-the-art Constraint Programming system Gecode (Gecode), the CPSP approach was reimplemented. The resulting CPSP-tools are a collection of programs related to lattice proteins in the HP model. Among them are the CPSP-implementation for optimal structure prediction HPstruct, tools for structure conversions and visualization (HPconv and HPview), or to calculate a sequence's degeneracy (HPdeg). The web-frontend CPSP-web-tools enables their online ad hoc usage. The generic Constraint Programming framework used within the CPSP approach enables further extensions to widen its applicability. Several such extensions have been discussed and were integrated into the CPSP-tools.

First, the CPSP extension to side chain models was introduced. The more detailed protein model reflects that only the side chain groups show a hydrophobic or hydrophilic behavior. Furthermore, the explicit representation of side chains avoids the formation of too compact structures as present in the backbone-only model. The CPSP approach for side chain models is still very efficient and fast despite the immensely larger structure space compared to backbone-only models. A first study revealed, that the increased structural flexibility in side chain models results in a very high degeneracy. Only about 4% of the sequences can adopt less than a million energy optimal structures.

Since most of the structures differ only in slightly changed side chain or backbone displacements, we were interested in a representative subset of structurally different conformations. Reconsidering that the energy function reflects only hydrophobic interactions, an energy induced equivalence relation was formulated that ignores backbone and polar side chain monomer placements. Thus, only if two structures differ in their corresponding relative hydrophobic monomer placement they are considered non-equivalent. The CPSP approach was extended in order to enumerate a representative structure for each resulting equivalence class. We denote the number of equivalence classes as core-degeneracy, since it corresponds to the number of differently formed hydrophobic cores the sequence can adopt. We found the core-degeneracy orders of magnitude smaller than degeneracy. In side chain models, all tested sequences show a core-degeneracy below one million while this holds only for 4% of the sequences when considering standard degeneracy. This shows the immense impact of non-constrained side chains and backbones on degeneracy.

The enumeration of structurally different representative structures opens the door to new applications of the CPSP approach. For instance, we could show that HP-optimal structures can be used to speed up local search procedures in more detailed protein models. A Simulated Annealing search protocol minimizing an empirical 20 amino acid energy function was shown to be more efficient starting from HP-optimal structures than from random conformations.

Backofen et al. (1999) have extended the CPSP approach to the more detailed HPNX energy model. The method is still based on hydrophobic cores as for the HP model but applies Constraint Optimization in order to minimize the HPNX energy function. The reimplementation of the method and its integration into the CPSP-tools collection would be useful. Furthermore, an extension of the approach to side chain models should be possible analog to the introduced CPSP-extension.

Over and above the application of more detailed energy models, further structural constraints could be considered in the CPSP approach. For instance, to constrain the torsion angles of the side chains to preferred values will drastically reduce the immense degeneracy in side chain models. On the other hand, the current approach is based on fixed H-cores and might be too rigid. The application of soft-constraints (*Bartak*, 2002) or limited discrepancy search (*Harvey and Ginsberg*, 1995) might be useful to formulate and resolve the problem.

CHAPTER IV

Folding Simulation in Side Chain Models

Publication

Constraint-based Local Move Definitions for Lattice Protein Models Including Side Chains

M. Mann*, A. Hamra, K. Steinhöfel, and R. Backofen*

in *Proc. of Workshop on Constraint Based Methods for Bioinformatics (WCB'09)*, 2009

arXiv:0910.3880

4.1 Introduction

The structure formation process of a protein after biosynthesis to attain its functional fold is called *protein folding*. Thus protein folding describes the protein's structural changes over time. A common visualization is a time series of structure snapshots, a *folding trajectory*, usually starting from a random coil conformation and ending in the functional fold. An example is given in Fig. 4.1.

A detailed high resolution study of protein folding *in vivo*, i.e. within living cells, is currently not possible. Therefore, computer simulations are used in order to create and evaluate folding trajectories *in silico*. According to Anfinsen's thermodynamic hypothesis, proteins fold into states

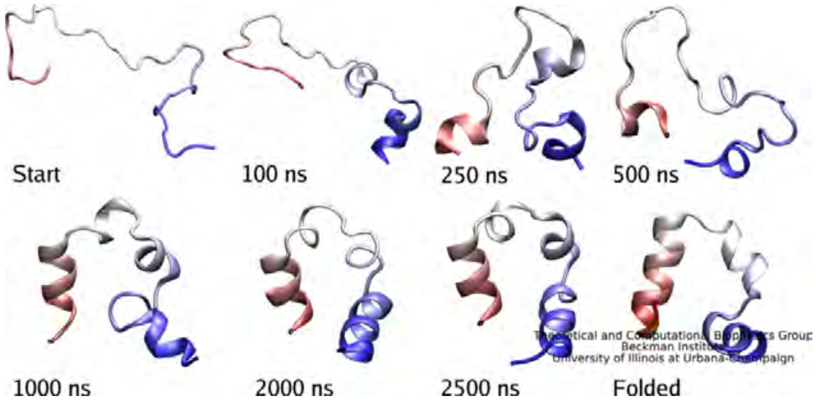

Start
100 ns
250 ns
500 ns
1000 ns
2000 ns
2500 ns
Folded
Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign

Figure 4.1: Folding trajectory of the fast folding WW domain of the villin protein. The image was created by *Beckman Institute at UIUC* (2008) using `VMD` (*Humphrey et al.*, 1996).

of minimal free energy such that their tertiary structure is encoded by the amino acid sequence (*Anfinsen*, 1973). To reduce the immense computational complexity, in general the folding of single molecules is simulated. Recently, *Jefferys et al.* (2010) have shown that this simplification might heavily bias the folding trajectories studied since crowding effects can reduce the flexibility of the folding process. Thus, some trajectories are ruled out or become more likely, leading to different final folds compared to single molecule folding. Furthermore, most experiments ignore the sequential biosynthesis of proteins during translation that can bias folding as well (*Kramer et al.*, 2009; *Ellis et al.*, 2010). The folding of the already produced parts of the protein during translation, so called co-translational folding, can guide the structure formation into the functional fold (*Huard et al.*, 2006). Also translation speed, temperature, the presence of helper molecules like chaperones or ligands, transportations events, or additional chemical changes are of importance for some proteins to attain their native structure (*Thanaraj and Argos*, 1996; *Clark*, 2004; *Ying et al.*, 2006).

Overall, the folding process underlies many external effects that make a full and detailed simulation very complicated and time consuming. Since thermodynamics are a major driving force of the process, most simulation protocols are studying thermodynamics-based (energy minimizing) folding trajectories only.

The most detailed studies of protein folding are done via Molecular Dynamics (MD) simulations (*Alder and Wainwright*, 1959). Here, the motions and interactions of all protein atoms and surrounding molecules like water are considered by approximations of the known physics and motion models of all elements. As discussed in Sec. 1.1, the thermodynamics of proteins is quite complex such that MD simulations are usually restricted to either very short time scales or short proteins (*Plimpton*, 1995). Thus, in order to enable larger folding studies more simplified models have to be considered. In the following, we will focus on folding protocols for lattice protein models.

Folding simulations are often based on stochastic local searches, e.g. Monte Carlo simulations introduced in Sec 1.4.2. Based on a neighborhood relation $\mathcal{N}$, local search traverses the structure search space in order to identify frequently used folding pathways, local/global energy minima, important intermediate structures, or other folding related properties. Different procedures, so called *move sets*, have been developed to calculate the structural changes along the simulation, i.e. to define and enumerate the structural neighborhood $\mathcal{N}$ of a certain structure. A method often applied in literature are *k-local moves* (*Sali et al.*, 1994) that allow for structural changes within a successive interval of fixed length k. They are discussed in detail in Sec 4.3.1. *Dotu et al.* (2008) have used local moves for backbone-only HP models within a constraint-based Large Neighborhood Search for optimal protein structures. Another important move set are the *pull moves* introduced by *Lesh et al.* (2003) for rectangular lattices that are widely used in recent studies, e.g. (*Swetnam and Allen*, 2010; *Ullah and Steinhöfel*, 2010). *Böckenhauer et al.* (2008) extended the work to triangular lattices. *Pivot moves* allow for the rotation or reflection of subchains at an arbitrary Pivot position of the structure (*Madras and Sokal*, 1988), while *Zhang et al.* (2007) suggested a sequential regrowth of structure fragments to enhance folding simulations.

All named move sets are currently restricted to backbone-only lattice protein models, i.e. only the C_α-trail of the protein is modeled. For more realistic protein models that incorporate side chains, often a combination of different move sets is applied. *Betancourt* (2005) combined Pivot moves on the backbone with a new so called `FEM` move set, while *Dima and Thirumalai* (2002) have used a combination of 2-local moves on the backbone with a simple relocation of the side chains. An exception is the advanced CABS model by *Kmiecik and Kolinski* (2007), which represents the side chain in higher detail and requires more complex moves.

Here, we introduce a generic and flexible approach to enable folding simulations in backbone-only and side chain models using any k-local moves (i.e. within any interval length k) in arbitrary lattices. The constraint programming (CP) based formulation focuses on a description of the targeted structural neighborhood $\mathcal{N}$ instead of an operational encoding of the moves possible. The introduced scheme is therefore easy to extend with new directives or can be used for other applications, e.g. fragment re-localization (*Zhang et al.*, 2007) as discussed later. Beneath applications in studies of the whole energy landscape (*Mann et al.*, 2007b), the approach is well placed to be applied within a local search following the framework of *Pesant and Gendreau* (1999).

Plan of the Chapter

Within Section 4.2, we repeat and give some preliminaries in oder to ease the presentation and understanding of the our k-local move definition for side chain models. The definition given afterwards is accompanied with a description of a constraint programming based framework that enables the direct enumeration of the according local neighborhood for a given structure (Sec. 4.3). We apply and test our move set in Sec. 4.4. To this end, a simple folding simulation procedure for side chain models is implemented that is similar to the approach presented in Sec. 3.5.

4.2 Preliminaries

In the following, some preliminaries and definitions from Sec. 1.2 are briefly repeated in order to ease the understanding of the approach introduced afterwards.

Given a lattice L, we describe a *backbone-only* lattice protein of length n by (S, P) where $S \in \Sigma^n$ denotes the sequence over some alphabet Σ and $P \in L^n$ the lattice nodes occupied by each amino acid monomer. A *side chain* lattice protein is defined by $(S, P{=}(P^b, P^s))$, i.e. it has backbone positions $P^b \in L^n$ and the side chain positions $P^s \in L^n$. The side chain positions P^s represent the centroids of the amino acids' side chain atoms where the backbone monomers P^b correspond to their C_α-positions. The constraints on a valid lattice protein structure are connectivity between successive monomers and its selfavoidingness, see Sec. 1.2.2.

We consider the contact based energy functions $E^b(S, P) = \sum_{1 \leq i+1 < j \leq n} e(S_i, P_i, S_j, P_j)$ for backbone-only (Eq. 1.19) and $E^s(S, (P^b, P^s){=}P) = \sum_{1 \leq i < j \leq n} e(S_i, P^s_i, S_j, P^s_j)$ for side chain lattice proteins (Eq. 1.20) for a given energy contribution function e. Here, e is a contact-based pairwise potential as defined in Eq. 1.22. In the following, e^{20} denotes an empirical 20×20 amino acid contact potential as described in (*Berrera et al.*, 2003; *Dal Palu et al.*, 2004). e^{HP} represents the energy contribution function of the Hydrophobic-Polar (HP) model (*Lau and Dill*, 1989) described in Sec. 1.2.3.1. Our hydrophobic/polar (H/P) assignment for amino acids follows *Ullah et al.* (2009) and is given in Tab. 1.3.

An *optimal structure* minimizes the used energy function. In the following, we denote a structure *HP-optimal* if it minimizes the energy function based on e^{HP}. Figure 4.2 exemplifies HP-optimal structures for both backbone-only and side chain lattice protein models. In the following, we assume a scaled lattice such that neighbored positions in the lattice have a distance of $3.8\mathring{A}$, the average C_α-atom distance in proteins.

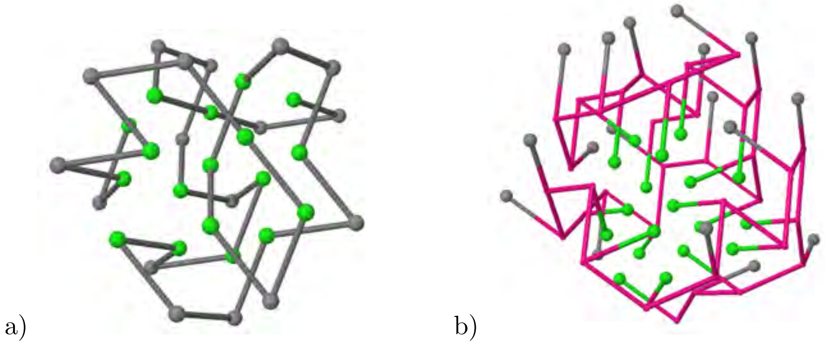


Figure 4.2: HP-optimal structures of `HPPHHPPPHPHHPHHPPHPHPPHHHPHHPPHPHPH` in the face-centered-cubic lattice. (a) backbone-only model with energy -50, (b) side chain model with energy -55. Colors: green - H-monomers, gray - P-monomers, red - backbone in side chain models. Calculation and visualization are done using the CPSP-package (*Mann et al.*, 2008b).

4.3 Local Moves for Side Chain Models

4.3.1 Constraint-based Neighborhood Generation

To enable folding simulations we need a definition of structural changes that encode the structural neighborhood $\mathcal{N}$ of a given lattice protein structure. Here, we follow the idea of k-local moves that confine the difference between the initial and the neighbored structure to a consecutive interval of a maximal length of k (*Sali et al.*, 1994). Therefore, we define the *k-neighborhood* $\mathcal{N}_k(P)$ of a given structure P of length n as:

$$\mathcal{N}_k(P) = \left\{ P' \mid \exists_{1\leq t\leq n} : \forall_{j\notin[t,\ldots,(t+k-1)]} : P_j = P'_j \right\} \tag{4.1}$$

The neighborhood definition $\mathcal{N}_k$ shows a recursive structure since it holds $\forall_{k'<k} : \mathcal{N}_{k'} \subseteq \mathcal{N}_k$. Thus, we can define the set of additional neighbored structures compared to $\mathcal{N}_{k-1}$, where the structural change really spans an interval of length k, as the *strict k-neighborhood* $\overline{\mathcal{N}}_k(P)$ given by

$$\begin{aligned} \overline{\mathcal{N}}_k(P) &= \mathcal{N}_k(P) \setminus \mathcal{N}_{k-1}(P) \\ &= \left\{ P' \mid \exists_{1\leq t\leq n} : P_t \neq P'_t \wedge P_{t+k-1} \neq P'_{t+k-1} \wedge \forall_{j\notin[t,\ldots,(t+k-1)]} : P_j = P'_j \right\} \ . \end{aligned} \tag{4.2}$$

Here, we enforce in addition that both ends (P'_t and P'_{t+k-1}) of the interval of length k are changed, i.e. a strict k-local move does not cover a k'-local move with $k' < k$, as a normal k-local move in accordance with Eq. 4.1 does.

One possible way to enumerate all valid structural neighbors $P' \in \mathcal{N}_k(P)$ of a given lattice protein P is to successively enumerate the strict k-neighborhoods for all possible smaller interval lengths $1 \leq k' \leq k$ since it holds

$$\mathcal{N}_k(P) = \bigcup_{1\leq k'\leq k} \overline{\mathcal{N}}_{k'}(P) \ . \tag{4.3}$$

It follows from Eq. 4.2 that $\forall_{1\leq k'<k''\leq k} : \overline{\mathcal{N}}_{k'}(P) \cap \overline{\mathcal{N}}_{k''}(P) = \emptyset$, since $\overline{\mathcal{N}}_{k''} = \mathcal{N}_{k''} \setminus \mathcal{N}_{k''-1}$ and

it holds that $\overline{\mathcal{N}}_{k'} \subseteq \mathcal{N}_{k''-1}$. Thus, every neighbored structure $P' \in \mathcal{N}_k(P)$ is part of only one strict neighborhood $\overline{\mathcal{N}}_{k'}(P)$. Thus, if we want to enumerate the k-local neighborhood $\mathcal{N}_k(P)$ of a given structure P, we can decompose this enumeration into k independent enumerations of the according strict neighborhoods $\overline{\mathcal{N}}_{k'}(P)$ for $1 \leq k' \leq k$. Due to the non-overlapping $\overline{\mathcal{N}}_{k'}(P)$, each structure is generated only once.

In the following, we will introduce the Constraint Satisfaction Problems (CSP) that describe all valid structural neighbors $P' \in \overline{\mathcal{N}}_k(P)$ of a given lattice protein P according to strict k-local moves in a lattice L. A CSP is given by $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where we denote the set of variables $\mathcal{X} = \{X_1, \ldots, X_n\}$, their according domains $\mathcal{D} = \{D_1, \ldots, D_n\}$, and a set of constraints $\mathcal{C}$. A solution of a CSP is an assignment $a_i \in D_i$ for each variable that satisfies all constraints in $\mathcal{C}$. To simplify the presentation, we utilize a binary lattice point neighboring constraint $\text{neigh}(X_i, X_j)$ that ensures $\forall_{d_i \in D_i} : \exists_{d_j \in D_j} : (d_i \in N_L(d_j))$ and vice versa. Furthermore, we use the global all-different constraint by *Régin* (1994) to enforce pairwise differences within a set of variables.

4.3.2 CSP for Backbone-only Models

Given a valid backbone-only lattice protein structure P of length n, a move interval length $k \leq n$, and the start of the interval $1 \leq t \leq (n - k + 1)$.

We define k variables X_i, one for each position of the interval. The domains of the variables are set to $D_i = L \setminus \{P_1, \ldots, P_{t-1}, P_{t+k}, \ldots, P_n\}$ such that we ensure self-avoidance with the remaining structure outside of the interval.

The variables have to form a valid (sub)structure. We enforce therefore pairwise difference using $\text{all-different}(X_1, \ldots, X_k)$ and chain connectivity via $\forall_{1 \leq i < k} : \text{neigh}(X_i, X_{i+1})$. Since we describe a substructure, it has to be connected to the interval borders: if $t > 1 : \text{neigh}(X_1, P_{t-1})$ and if $(t + k - 1) < n : \text{neigh}(X_k, P_{t+k})$. Finally, we enforce that both ends of the interval are different from the old placement, i.e. $X_1 \neq P_i$ and $X_k \neq P_{i+k-1}$, to enumerate strict k-local move neighbors only.

The presented CSP is similar to the work of *Dotu et al.* (2008) but in contrast ensures the uniqueness of each move. Thus, each neighbored structure is available only via a single interval. This is of high importance to enable a non-redundant enumeration of a structure's neighborhood in the fold space to access its energy landscape (*Mann et al.*, 2007b). Figure 4.3 exemplifies 1-local moves in the 2D-square lattice.

4.3.3 CSP for Side Chain Models

Given a valid side chain lattice protein structure (P^b, P^s) of length n, a move interval length $k \leq n$, and the start of the interval $1 \leq t \leq (n - k + 1)$.

We define $2k$ variables X^b_i and X^s_i, two for each position of the interval. The initial domains are given by $D^b_i = D^s_i = L \setminus \{P^b_1, \ldots, P^b_{t-1}, P^b_{t+k}, \ldots, P^b_n, P^s_1, \ldots, P^s_{t-1}, P^s_{t+k}, \ldots, P^s_n\}$. This initialization ensures the avoidance of the remaining structure outside of the interval.

To ensure a valid structure, we enforce self-avoidance by $\text{all-different}(X^b_1, \ldots, X^b_k, X^s_1, \ldots, X^s_k)$, backbone connectivity with $\forall_{1 \leq i < k} : \text{neigh}(X^b_i, X^b_{i+1})$, and side chain connectivity using $\forall_{1 \leq i \leq k} : \text{neigh}(X^b_i, X^s_i)$. Since we describe a substructure, the backbone has to be connected to the outside of the interval borders: if $t > 1 : \text{neigh}(X^b_1, P^b_{t-1})$ and if $(t+k-1) < n : \text{neigh}(X^b_k, P^b_{t+k})$. Finally, we

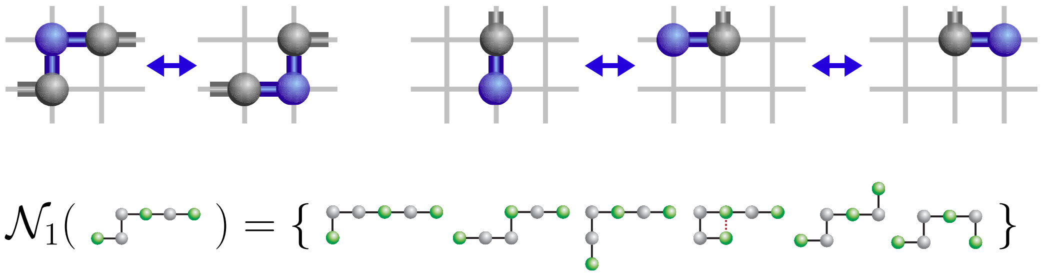


Figure 4.3: In the upper part, all possible 1-local moves for backbone-only models in the 2D-square lattice are given, i.e. on the upper left a "corner-flip" move and on the upper right all possible end moves. Below, the according 1-local move neighborhood $\mathcal{N}_1$ of the left structure is presented.

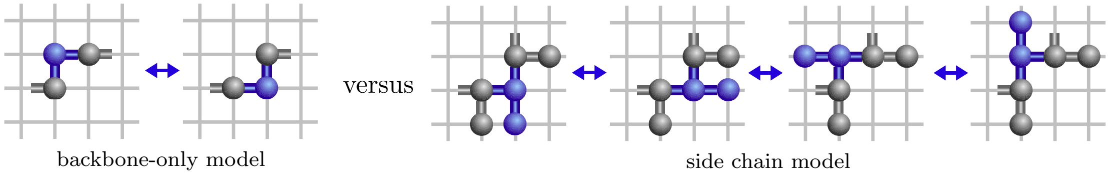


Figure 4.4: 1-local moves based on a "corner-flip" (left) in backbone-only models and (right) in side chain models. Note, the structural flexibility in side chain models is much higher than in backbone-only models.

warrant the strictness of the k-local moves and enforce that both ends of the interval differ from the old backbone or side chain placement, i.e. $(X_1^b \neq P_i^b \vee X_1^s \neq P_i^s)$ and $(X_k^b \neq P_{i+k-1}^b \vee X_k^s \neq P_{i+k-1}^s)$.

This CSP enables the enumeration of the strict k-local move neighborhood $\overline{\mathcal{N}}_k$ of a side chain structure in arbitrary lattice protein models. Figure 4.4 shows some 1-local moves for side chain structures related to the "corner-flip" move in backbone-only models. Note, the additional flexibility in side chain positioning increases the number of possible moves drastically. This is even more severe in more complex 3D lattices.

4.4 Application and Results

In the following, we describe the application of the introduced move set to folding simulations of side chain lattice protein models in the 3D face-centered-cubic (FCC) lattice (defined in Sec. 1.2.1). k-local moves are known to be non-ergodic for backbone-only models (*Madras and Sokal*, 1987) depending on the interval length k, the used lattice, and the protein length. We expect the same for models including side chains, but using the FCC and an intermediate k should shift the problem to long chain lengths not to be considered in this study. Thus, we apply 3-local moves, i.e. with a maximal interval length $k = 3$ such that up to 6 monomers are moved (2 per sequence position). The implementation is based on the C++ Gecode Constraint Programming framework (Gecode).

To evaluate the structural difference between two structures $P = (P^b, P^s)$ and $\hat{P} = (\hat{P}^b, \hat{P}^s)$ we calculate the distance and coordinate root mean square deviation (dRMSD and cRMSD) as given by Eq. 2.5 and 2.3, respectively. The needed superpositioning utilizes Kabsch's algorithm (*Kabsch*, 1976, 1978). We apply the contact based energy function E^s that evaluates (only) side

PDB ID - chain	Sequences (original and HP transform)
1BAZ-A	SKMPQVNLRWPREVLDLVRKVAEENGRSVNSEIYQRVMESFKKEGRIGA PPHPPHPHPHPPPHHPHHPPHPPPPPPPHPPPHHPPHHPPHPPPPPHPP
1J8E-A	GSHSCSSTQFKCNSGRCIPEHWTCDGDNDCGDYSDETHANCTNQ PPPPHPPPPHPHPPPPHHPPPHPHPPPPPHPPHPPPPPPPPHPPP
1RH6-A	MYLTLQEWNARQRRPRSLETVRRWVRESRIFPPPVKDGREYLFHESAVKVDLNRP HHHPHPPHPPPPPPPPPHPPHPPHHPPPPHHPPPHPPPPPHHHPPPPHPHPHPPP
1Z0J-B	IEEELLLQQIDNIKAYIFDAKQCGRLDEVEVLTENLRELKHTLAKQKGGTD HPPPHHHPPHPPHPPHHHPPPPHPPHPPHPHHPPPHPPHPPPHPPPPPPPP
2DS5-A	GKLLYCSFCGKSQHEVRKLIAGPSVYICDECVDLCNDIIREEI PPHHHHPHHPPPPPPHPPHHPPPPHHHHPPHHPHHPPHHPPPH
2EQ7-C	LAMPAAERLMQEKGVSPAEVQGTGLGGRILKEDVMRH HPHPPPPPHHPPPPHPPPPHPPPPHPPPHHPPPHHPP
2HBA-A	MKVIFLKDVKGMGKKGEIKNVADGYANNFLFKQGLAIEATPANLKALEAQKQ HPHHHHPPHPPHPPPPPPHPPHPPPHPPPHHHPPPHPHPPPPPPPHPPHPPPPP

Table 4.1: Used protein sequences and their HP transforms according to Tab. 1.3.

PDB ID		P_{fit} to P_{PDB}				$gw(P_{\text{fit}})$ to P_{fit}	
- chain	n	dRMSD	cRMSD	$E(P_{\text{fit}})$	$E(gw(P_{\text{fit}}))$	dRMSD	cRMSD
1BAZ-A	49	0.886 Å	1.725 Å	-3.73	-31.51	4.050 Å	6.565 Å
1J8E-A	44	0.928 Å	1.939 Å	-3.54	-30.76	3.865 Å	6.857 Å
1RH6-A	55	0.921 Å	1.791 Å	1.33	-38.17	4.192 Å	8.243 Å
1Z0J-B	51	0.917 Å	2.095 Å	2.05	-35.95	3.185 Å	6.640 Å
2DS5-A	43	0.901 Å	1.750 Å	-4.35	-34.36	4.658 Å	7.755 Å
2EQ7-C	37	0.905 Å	1.813 Å	-3.07	-20.58	2.328 Å	4.751 Å
2HBA-A	52	0.890 Å	1.780 Å	-3.04	-30.62	3.224 Å	6.015 Å

Table 4.2: The quality of the fitted lattice protein models $P_{\text{fit}}/gw(P_{\text{fit}})$ compared to the original proteins P_{PDB} from Tab. 4.1 and the according energies of the model structures.

chain monomer contacts using the e^{20} contact energy potentials as described in Sec. 4.2 in accordance to the backbone-only studies by *Berrera et al.* (2003) and *Ullah et al.* (2009).

We derived real protein data from the Pisces web server (*Wang and Dunbrack*, 2003) on June 23rd 2009. Only complete X-ray structures of 2.0Å resolution or better with an R-value[1] of 0.3 that contain side-chain data were considered. We used a 30% sequence identity cut-off. Since we are applying a simple contact-based energy function we filtered for short globular shaped proteins. Table 4.1 summarizes the used sequences and their corresponding Protein Data Bank (PDB) identifiers etc.

For each full atom PDB structure P_{PDB}, we derived a lattice protein structure P_{fit} that minimizes the dRMSD to P_{PDB}. This was done using LatFit introduced in Sec. 2.2 from the LatPack-tools package v1.7.0 (*Mann et al.*, 2008a). Table 4.2 summarizes the resulting dRMSD and cRMSD values (column "P_{fit} to P_{PDB}").

[1] The R-value measures the quality of the atomic structure model via a comparison to simulated data and is provided within the PDB protein database. A totally random model will give an R-value of about 0.63, whereas a perfect model will produce a value of 0.

Since the applied energy function is still a rough abstraction of the forces that guide the real folding process into P_{PDB}, no energy minimizing folding strategy will find the fitted lattice protein structure P_{fit}. Thus, we map P_{fit} to the according local minimum in the energy landscape. The mapping is done via a steepest decent or *gradient walk*. Starting from a given structure, at each step the neighbored structure with lowest energy is chosen for the next step until no such neighbor exists. Therefore, a gradient walk ends in a local minimum of the energy landscape, which we denote $gw(P)$ for a given start structure P.

The $gw(P_{\text{fit}})$ structures represent our "true" model to benchmark the following folding scheme. The energies of P_{fit} and $gw(P_{\text{fit}})$ and their structural differences to each other and to P_{PDB} are given in Table 4.2. The gradient walks reveal that the currently applied contact based energy function using the energy potentials e^{20}, originally derived for backbone-only models (*Berrera et al.*, 2003), does not reflect the forces needed in models including side chains. This can be observed when comparing the energies $E(P_{\text{fit}})$ to $E(gw(P_{\text{fit}}))$ in Table 4.2. The energy of the fit P_{fit} is much higher when compared to the energy of the according local minimum $gw(P_{\text{fit}})$. An energy function that results in a smaller difference would be preferable if we want to find the fitted structure via folding simulations in the lattice model. Therefore, we can only target the local minimum assigned via $gw(P_{\text{fit}})$ within our folding simulations.

The folding simulation procedure applied follows the idea presented in (*Ullah et al.*, 2009). For each amino acid sequence S, we derive an according HP-sequence S_{HP} using the translation table from Tab. 1.3. The derived sequences are given in Table 4.1. Following the observation of the hydrophobic collapse (*Agashe et al.*, 1995), we calculated HP-optimal structure representatives utilizing the CPSP approach (*Backofen and Will*, 2006; *Mann et al.*, 2008b, 2009c) and its latest extension HPREP (*Mann et al.*, 2009a) described in Sec. 3.4. The resulting HP-optimal structures are named P_{HP}. For each P_{HP} we run gradient walks and evaluated the resulting local minima found. The corresponding energies are listed in Table 4.3. Furthermore, we performed a structural comparison of the resulting $gw(P_{\text{HP}})$ structures to our "true" models $gw(P_{\text{fit}})$ from the fitting. The RMSD values are given in Table 4.3.

In addition, we executed for each P_{HP} *random descending walks* in order to sample the local minima of the energy landscape accessible from the collapsed starting structures. Here, at each step a random neighbor with lower energy is selected following a uniform distribution until no such neighbor exists. The lowest reached local minimum of all random descending walks starting at P_{HP} is denoted by $rw(P_{\text{HP}})$. Energy and structural differences are given in Table 4.3.

We see from the data in Table 4.3 that the derived structures from our simple energy-optimizing folding simulation procedure are still quite dissimilar to the energy-optimized lattice fits of the real structures. We assume that this mainly results from the simple energy function as well.

4.5 Summary and Future Work

We introduced a CP-based approach to enumerate k-local neighbors of lattice protein structures in backbone-only and side chain lattice protein models. The generic approach can be applied for any local move length k within arbitrary lattices. Thus, it enables a fast prototyping of new folding simulation schemes or can be easily extended with additional constraints, e.g. restricted torsion angles.

PDB ID	average values	minimal values	
- chain	$\langle E(P_{\mathrm{HP}})\rangle$	$\min E(gw(P_{\mathrm{HP}}))$	$\min E(rw(P_{\mathrm{HP}}))$
1BAZ-A	-10.67	-33.07	-34.60
1J8E-A	-12.45	-29.33	-32.35
1RH6-A	-13.09	-35.12	-37.59
1Z0J-B	-13.42	-34.71	-37.69
2DS5-A	-6.97	-31.00	-32.53
2EQ7-C	-6.55	-21.64	-25.10
2HBA-A	-11.07	-30.91	-35.56

PDB ID	$gw(P_{\mathrm{HP}})$ vs. $gw(P_{\mathrm{fit}})$		$rw(P_{\mathrm{HP}})$ vs. $gw(P_{\mathrm{fit}})$	
- chain	dRMSD	cRMSD	dRMSD	cRMSD
1BAZ-A	4.736 $\mathring{A}$	8.797 $\mathring{A}$	4.762 $\mathring{A}$	9.360 $\mathring{A}$
1J8E-A	3.384 $\mathring{A}$	7.508 $\mathring{A}$	3.196 $\mathring{A}$	7.052 $\mathring{A}$
1RH6-A	4.190 $\mathring{A}$	9.645 $\mathring{A}$	4.242 $\mathring{A}$	10.156 $\mathring{A}$
1Z0J-B	5.609 $\mathring{A}$	10.166 $\mathring{A}$	6.232 $\mathring{A}$	11.438 $\mathring{A}$
2DS5-A	3.588 $\mathring{A}$	8.679 $\mathring{A}$	3.425 $\mathring{A}$	7.639 $\mathring{A}$
2EQ7-C	3.427 $\mathring{A}$	7.247 $\mathring{A}$	4.177 $\mathring{A}$	8.401 $\mathring{A}$
2HBA-A	3.832 $\mathring{A}$	8.848 $\mathring{A}$	4.194 $\mathring{A}$	9.075 $\mathring{A}$

Table 4.3: Resulting energies and a structural comparison of the folding results.

The CSP formulation enables the enumeration of the whole k-local move neighborhood $\mathcal{N}_k(P)$ of a given structure P or the calculation of a random neighboring structure $P' \in \mathcal{N}_k(P)$ when applying a randomized search as possible in Gecode (Gecode). The application of symmetry breaking search (*Backofen and Will*, 2002) can be used to avoid the enumeration of symmetric structures, while the latter increases the efficiency of folding simulations as shown by *Gan et al.* (2008). We plan the incorporation of the k-local move neighborhood enumeration into our C++ energy landscape library (ELL) (*Mann et al.*, 2007b). This will open an easy interface for folding simulations in arbitrary lattices utilizing any energy function of interest. Furthermore, this will enable full kinetics studies based on the energy landscape topology.

In addition, one can use the flexibility of the CP-based approach to incorporate additional structural constraints into the neighborhood generation. For instance it was shown by *Dal Palu et al.* (2004) and *Dal Palù et al.* (2004) that it is beneficial to restrict torsion angles along the backbone or to exploit secondary structure information. Both can be incorporated into our framework.

Another advantage of the CP-based approach is its extensibility to constraint optimization problems (COP). Currently, we plan to incorporate the energy function as the objective into the CSP, as e.g. done in (*Cipriano et al.*, 2009; *Dotu et al.*, 2008). Thus, by solving a COP while optimizing the energy function, we can directly calculate the lowest energy neighbor of a structure following the framework of *Pesant and Gendreau* (1999). This is needed e.g. for a gradient walk in the energy landscape as done in Sec. 4.4. Furthermore, this would enable an extension of the work of *Zhang et al.* (2007). They showed (for backbone-only models) that the performance of Monte Carlo folding simulations can be significantly increased using a greedy sequential regrowth of subchains. Thus, we plan to directly apply the sketched COP to calculate the optimal fragments for lattice proteins including side chains. Finally, the presented CP-based move set formulation can be easily extended to any other local move definition of interest.

CHAPTER V

Co-translational Folding

Publication

Signatures of Co-translational Folding

R. Saunders, **M. Mann***, *and C. Deane*

in *Biotechnology Journal, Special Issue on Protein Folding in vivo*, 2011, vol. 6(6).

* These authors contributed equally to this work.

doi:10.1002/biot.201000330

5.1 Introduction

Whether protein structure formation occurs concurrently with translation or after release from the ribosome is currently widely debated, see reviews (*Kramer et al.*, 2009; *Cabrita et al.*, 2010). There is experimental evidence for both folding mechanisms (*Anfinsen et al.*, 1961; *Sanchez et al.*, 2004; *Kolb et al.*, 2000; *Nicola et al.*, 1999; *Fedorov and Baldwin*, 1997). If both occur *in vivo* then understanding the features, interplay and relative abundance of these two folding mechanisms may play a key role in furthering our knowledge of protein structure.

A protein has a huge number of possible structures but, in accordance with the Levinthal paradox, it cannot explore every possible conformation in finding its biological structure (*Levinthal*, 1968). It has been suggested that a directed pathway through structure space is used to attain the biological structure (*Karplus*, 1997). Under *co-translational (CT) folding*, the vectorial emergence of proteins from the ribosome could form the basis for such directed folding pathways.

Experiments have demonstrated that ribosomes can catalyse the folding process (*Hardesty and Kramer*, 2001; *Kramer et al.*, 2009; *Saunders and Deane*, 2010) and simulations have suggested possible mechanisms for this effect. A simulation on the diamond lattice by *Sikorski and Skolnick* (1990) found that the ribosome accelerated the folding process by preventing formation of off-pathway intermediates. A model by *Contreras Martinez et al.* (2006) found that the ribosome's exit tunnel can facilitate folding of well-designed proteins. By preventing certain sub-structures in analogy to the exit tunnel restrictions, *Chikenji et al.* (2006) transformed the normally rugged energy landscape of simplified models into a smooth funnel. Recently *Jefferys et al.* (2010) used a CT protein folding algorithm to study the importance of macromolecular crowding on protein folding while *Ellis et al.* (2010) proved that the translation direction has a strong impact on protein folding.

Plan of the Chapter

Within this chapter CT folding is studied using the HP model. In Sec. 5.2, a stochastic simulation scheme for CT folding is introduced and verified. It is used to evaluate a classification protocol for the identification of sequences that benefit from CT folding (Sec. 5.3.1). The classification procedure enables the generation of large positive and negative data sets to study the implications of CT folding. Such a large scale classification is not possible with an explicit CT folding simulation. Within Section 5.4, we study the signatures in sequence and structure space that are specific to CT folding proteins. The study shows that CT folding can serve as a robust way for many proteins to attain their functional fold. Therefore, we introduce a folding based scheme to identify protein-like sequences in the HP model that accounts for CT folding (Sec. 5.5).

5.2 Simulating Co-translational Folding

Translation speed affects protein folding (*Zhang et al.*, 2009) and we hypothesise that folding during elongation can restrict the final conformation set produced. Furthermore, it is know that folding occurs orders of magnitude faster than elongation (*Basharov*, 2000). Thus, a CT folding simulation should incorporate such *interim folding* of the already extruded polypeptide during chain elongation (*Skolnick and Kolinski*, 1991).

Here, we introduce a CT folding scheme to investigate this principle and we term the resulting conformation a *CT-fold*. To model interim folding, a Markov-chain simulation is used after each elongation event. This provides an energy-biased exploration of the local conformation space during elongation. In Sec. 5.2.1, we introduce our CT folding simulation protocol and evaluate it in Sec. 5.2.2.

5.2.1 Method

For a given protein sequence S of length n, the proposed CT folding process uses an iterative scheme of $n-1$ elongation events, each followed by interim folding. The phases in detail are:

Elongation: Given the final conformation P of length $l-1$ of the last iteration. We produce the set of all its possible elongations $\mathbb{P}^l$ of length l by adding one monomer at the growing end of the structure P. The number of such elongations is restricted by the number of neighboring vectors in the lattice, i.e. $|N_L|$. From the resulting conformation ensemble $\mathbb{P}^l$, we pick a random elongation $P' \in \mathbb{P}^l$ according to its Boltzmann probability within the ensemble given by

$$Pr(P' \in \mathbb{P}^l) = \frac{\exp(-E(P')/kT)}{\sum_{P^* \in \mathbb{P}^l} \exp(-E(P^*)/kT)} \,, \tag{5.1}$$

at a given relative temperature kT (see Sec. 1.4.2.1). The term $E(P')$ abbreviates the energy function $E([S_1, \ldots, S_l], P')$ for the structure P' representing the according subsequence of length $l \leq n$ (see Sec. 1.2.3).

Interim folding: The chosen elongation P' defines the start conformation for interim folding. Interim folding is simulated via a Markov-chain protocol using the Metropolis algorithm described in Sec. 1.4.2.1. The available state space $\mathbb{Q}$ is defined by the set of all structures of length l, i.e. it includes P' and grows exponentially in l (see Sec. 1.2.2). The possible transitions are defined by *pull-moves* introduced by *Lesh et al.* (2003) where the probabilities for transition from P_{from} to P_{to} are given by the Metropolis criterion $\min\left\{ 1 \,,\, e^{-(E(P_{\text{to}})-E(P_{\text{from}}))/kT} \right\}$, introduced in detail in Eq. 1.33 on page 25.

We start the interim folding with structure P' and explore a length l dependent number of folding events $c{\cdot}l$ using a constant c. The length-dependent folding time $(c{\cdot}l)$ incorporates the locality of single *pull-moves*. Thus, c is the average number of folding events per residue. This is necessary because the simulation applies a series of local structural changes that, in biology, could occur simultaneously.

The final conformation resulting from interim folding either starts another elongation iteration (if $l < n$) or is a *CT-fold* of the sequence. In general, we envision that CT folding produces a limited set of structures (CT-folds) that provide a beneficial start point for reaching the biological conformation. One CT-fold might, of course, be the biological conformation.

A C++ implementation of the method, named LatFoldVec, is available within the LatPack-package v1.8.0 (*Mann et al.*, 2008a). For global folding we use our implementation LatFold from the same package.

5.2.2 Evaluation

To validate our method of producing CT-folds we investigate the 10-mer sequence `HPHPPHPPHH` taken from (*Huard et al.*, 2006) within the 2D square lattice. The sequence can form only one structure with minimal energy, defined to be its functional fold. *Huard et al.* (2006) have demonstrated, via a full enumeration of all possible CT folding paths, that this sequence folds more efficiently co-translationally than globally. In the following we can show that the sequence also shows this trend using our folding procedure. Based on previous work (*Mann et al.*, 2008a; *Jacob and Unger*, 2007) we use $kT = 0.3$ as the relative folding temperature within the Metropolis criterion and to choose and elongation in Eq. 5.1. Other values of kT (0.1 to 0.5) produced similar results.

As noted above, we do not assume that the functional (presumed energy minimal) structure is always attained directly through CT folding. Rather that CT folding benefits the protein by promoting the formation of the biological structure. For instance, in nature many proteins are active immediately after, but not before, release from the ribosome (*Hamlin and Zabin*, 1972; *Nicola et al.*, 1999; *Kolb et al.*, 2000; *Sanchez et al.*, 2004). Here, we envision that CT folding produces a limited set of structures that provide a beneficial start point for attaining the biological conformation. Thus, we apply a two step scheme to mimic nature: first we apply the CT folding simulation producing a CT-fold. This fold starts a short global folding simulation for the full length protein to give it further time to attain its functional fold. Afterwards, we test if the functional fold was adopted or not.

In detail, different intermediate folding times (c constant) between elongation events are tested; with four test sets of CT-folds produced ($c = 0, 1, 2, 3$). The success rate of attaining the functional fold from CT-folds is compared to that of attaining the functional fold from a set of random start conformations. In each case, 10^5 global folding simulations of 100, 200, and 500 simulation steps are undertaken. Global folding is simulated by the Metropolis algorithm from Sec. 1.4.2.1 using the same parametrization as for the interim folding within the CT folding simulation but for full length structures. The protein's *folding rate* r_f is calculated from the number of successful (the functional structure reaching) folding simulations h_{succ} in the 10^5 simulations: $r_f = (h_{succ}/10^5)$.

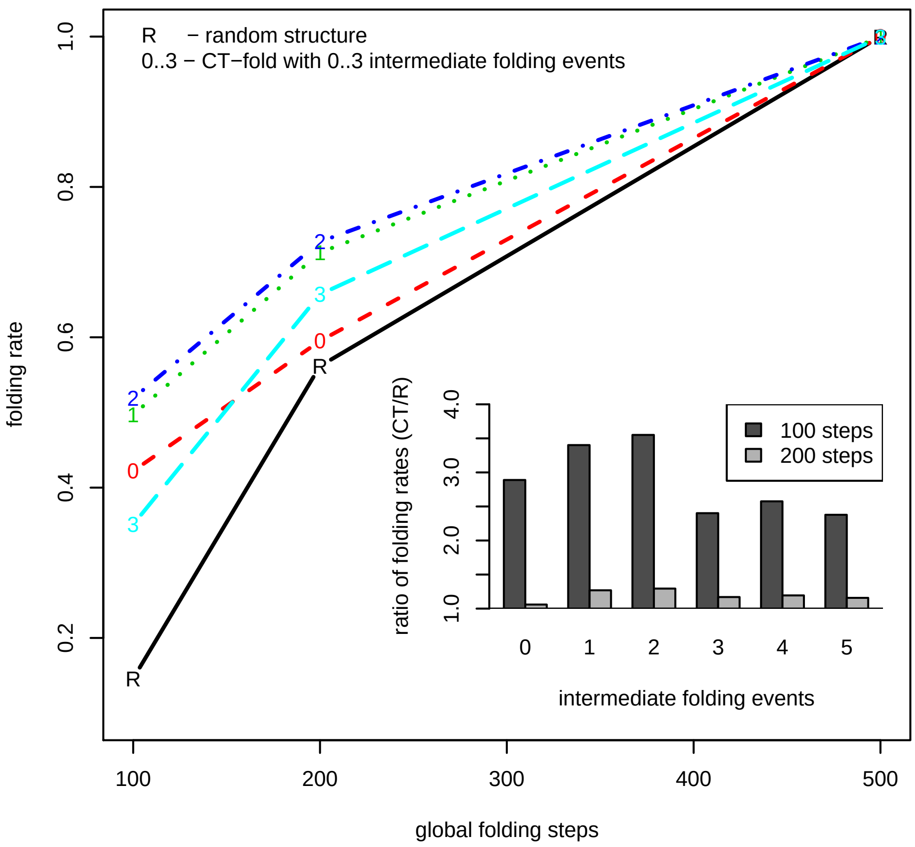


Figure 5.1: The effect of start conformations on attaining the functional fold of sequence `HPHPPHPPHH`. The proportion of simulations resulting in the energy minimal conformation (Y-axis) is plotted against the number of global folding steps (X-axis). Starting from randomly compact structures (R, solid line) is never better than starting from a CT-fold. Data is shown for CT-folds created using $c = 0, 1, 2,$ and 3 intermediate folding events per elongation (dashed lines). The inset chart shows the relative success rate (Y-axis) of using CT-folds over randomly compact structures as a start point for CT-folds created using a varying number of intermediate folding events (X-axis).

Figure 5.1 summarises the results. Our model produces the folding behaviour for `HPHPPHPPHH` as predicted by full folding pathway enumeration (*Huard et al.*, 2006). Starting from CT-folds is always superior to random start conformations for this sequence. As the number of folding events per elongation is increased between 0 and 2, the folding rate also increases. Above two intermediate folding events no improvement in the folding rate is observed for this sequence. This may result from the 1st-order Markov-chain implemented because as time increases the protein chain becomes increasingly independent of the start conformation. These results demonstrate that the model can reproduce the exhaustive study and that model proteins can benefit from CT folding.

5.3 Identifying Co-translationally Folding Sequences

Studies of co-translational (CT) folding require for data sets of proteins that fold co-translationally and ones that do not benefit from CT folding. Based on such data one can study the differences between the data sets in order to identify the features in proteins' sequence and structure space that are related to CT folding. Within this section, a classification scheme is introduced that enables a fast generation of such data sets. After its evaluation, the procedure is applied in the following section (Sec. 5.4) in order to identify signatures of CT folding in proteins.

Generally, only HP sequences with a unique global energy minimum (UGEM) conformation are considered protein-like. Thus, taking only sequences with a UGEM structures we want to produce three sequence sets for our studies:

1. *Global-CT* - sequences reaching their UGEM conformation co-translationally.
2. *Kinetic-CT* - a final unique conformation is formed co-translationally but it is not the UGEM conformation.
3. *Non-CT* - no unique final CT conformation is attained, i.e. the sequence does not benefit from CT folding.

The Kinetic-CT set is included because previous research has shown that the UGEM conformation is not always the most highly populated state under CT folding (*Huard et al.*, 2006) whereby CT folding can produce stable but thermodynamically non-optimal structures (*Morrissey et al.*, 2004).

CT folding simulation protocols similar to the one presented in Sec. 5.2 have been informative (*Skolnick and Kolinski*, 1991; *Bromberg and Dill*, 1994; *Wang and Klimov*, 2007) but are to slow to be applied to high-throughput screens for the classification at hand. Therefore, we introduce and apply a different method based on a chain-growth procedure similar to (*Bornberg-Bauer*, 1997a). The method is deterministic and fast showing a linear runtime in sequence length. In the following Sec. 5.3.1, we introduce the classification procedure and evaluate it in Sec. 5.3.2.

5.3.1 Method

Our three sequence sets are created using the following classification scheme that evaluates folding via low-energy pathways available within the complete energy landscape. CT folding is modeled by a chain-growth procedure , which we apply in accordance with *Huard et al.* (2006). For each elongation event $l \in [1, n]$, all prior conformations are extended to generate $\mathbb{P}^l$, a set of conformations of length l. Only conformations that show an energy within ΔE of the current minimum energy within $\mathbb{P}^l$ are accepted; thus ΔE is the energy in the system available for refolding (*Huard et al.*, 2006). At full elongation length n, we investigate the energy minimal conformations within $\mathbb{P}^n$ resulting from CT folding. Based on that we classify the according sequence as either Global-CT, Kinetic-CT, or Non-CT.

More precisely, for a given protein sequence S of length n we follow an iterative scheme of $n-1$ elongation events followed by a pruning step. In the following, the set of structures of length l considered for elongation is named $\mathbb{P}^l$. We start with a structure of one monomer representing S_1, i.e. the structure set $\mathbb{P}^1$. For each elongation event l ($2 \leq l \leq n$), all conformations from the last elongation $P \in \mathbb{P}^{l-1}$ are extended to produce all possible elongated conformations $\mathbb{P}^l_{\text{all}}$ of length l. Since we are only interested in low-energy folding pathways, we have to filter the set of all possible elongations $\mathbb{P}^l_{\text{all}}$ to obtain the set of structures $\mathbb{P}^l$ to be considered in the next iteration. For that we identify the minimal energy E_m of all $P' \in \mathbb{P}^l_{\text{all}}$ calculated by $E_m(\mathbb{P}^l_{\text{all}}) = \min(E(P')\ |P' \in \mathbb{P}^l_{\text{all}})$. The term $E(P')$ abbreviates the energy function $E([S_1,\dots,S_l],P')$ for the structure P' representing the according subsequence of length $l \leq n$ (see Sec. 1.2.3). Only conformations $\check{P} \in \mathbb{P}^l_{\text{all}}$ within ΔE of E_m are retained for extension at the next elongation event, such that

$$\mathbb{P}^l = \{\check{P} \mid \check{P} \in \mathbb{P}^l_{\text{all}} \wedge (E(\check{P}) \leq (E_m(\mathbb{P}^l_{\text{all}}) + \Delta E))\}.$$

The final set $\mathbb{P}^n$ is restricted to the minimal energy conformations reachable, i.e.

$$\mathbb{P}^n = \{\check{P} \mid \check{P} \in \mathbb{P}^n_{\text{all}} \wedge E(\check{P}) = E_m(\mathbb{P}^n_{\text{all}})\}.$$

Given that the protein with sequence S has a UGEM conformation of energy $E(\textsc{ugem})$ we can classify S according to the minimal energy within the structure set $\mathbb{P}^n$ and its size:

- S is *Global-CT* if the UGEM structure was found,
  i.e. $E(\textsc{ugem}) = E_m(\mathbb{P}^n)$; from this it follows that $|\mathbb{P}^n| = 1$,
- or *Kinetic-CT* if a single energy minimal structure but not the UGEM structure was obtained,
  i.e. $|\mathbb{P}^n| = 1 \wedge E(\textsc{ugem}) \neq E_m(\mathbb{P}^n)$,
- or *Non-CT* if no unique structure was reached,
  i.e. $|\mathbb{P}^n| > 1$.

A C++ implementation of the method, named LatVec, is available within the LatPack-package v1.7.0 (*Mann et al.*, 2008a).

5.3.2 Evaluation

Using our classification method, we have classified HP sequences into three sets: Global-CT, Kinetic-CT, and Non-CT folders. The 765, 147 sequences of length 25 with a UGEM conformation in the 2D-square lattice (*Irbäck and Sandelin*, 2000) are separated into 17, 085 Global-CT, 74, 502 Kinetic-CT, and 673, 560 Non-CT sequences. We used a surmountable energy barrier of zero ($\Delta E = 0$) that prevented any chance of global folding.

Our classification procedure has no kinetic component. It is a purely deterministic measure of whether a sequence can benefit from CT folding under the most restrictive definition of CT folding. To evaluate our classification procedure we investigate the folding rates of a random subset of each group (Global-CT, Kinetic-CT, Non-CT) using the Markov-chain CT folding simulation (Sec. 5.2) followed by global folding (as done in Sec. 5.2.2) using $kT = 0.3$. It is expected that Global-CT sequences will exhibit the highest success rate when global folding is initiated from CT folds. Furthermore, Kinetic-CT sequences should have a preference for reaching their kinetic fold over their UGEM conformation.

Figure 5.2 shows the ratio of the folding rates when starting from CT-folds compared to starting from random conformations. Folding rates are averaged over 50 randomly selected sequences per group and 10^4 folding simulations of 200 global folding steps per sequence and number of intermediate folding events (c). As expected, Global-CT sequences benefit most from CT folding and show a higher folding rate increase than Non-CT sequences. In reaching their UGEM conformation, Kinetic-CT sequences show a similar folding rate increase for increasing*c compared to Non-CT sequences *but* show a significantly higher rate to adopt their proposed kinetic fold. For all classifications, starting from the CT-folds increases the folding rate and we suggest that CT folding is an efficient way to explore fold space. Increasing the number of intermediate folding events per elongation (c) is beneficial up to a particular point. The optimal number of intermediate folding events increases with sequence length, but we elucidate no general rule underlying it. The effect probably results from a longer energy driven exploration of CT structure space.

Overall, our extensive sequence classification is supported by the differing folding properties of our three sets under a Markov-chain simulation.

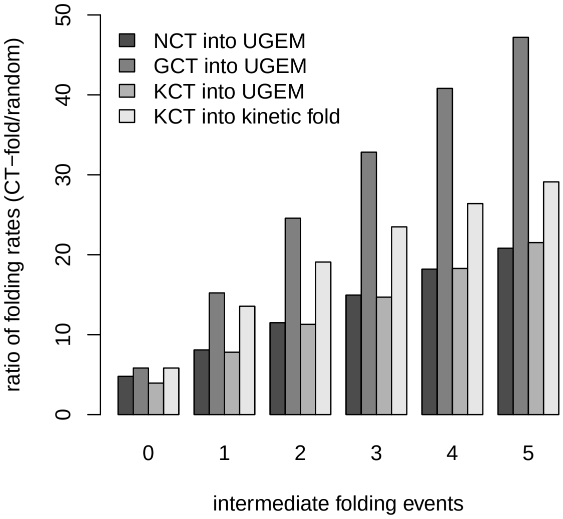


Figure 5.2: The propensity of our different sets to fold successfully. The ratio of the average folding rates (Y-axis) is the success rate when starting from CT-folds compared to the success rate starting from random structures. Data is shown for different numbers of intermediate folding events (parameter c, X-axis) using protein-like sequences of length 25 in the 2D-square lattice. Data for our Kinetic-CT set is split into folding to the UGEM conformation and to the unique kinetic fold identified by our classification system. NCT = Non-CT folders, GCT = Global-CT and KCT = Kinetic-CT.

5.4 Properties of Co-translationally Folding Proteins in HP-models

CT folding occurs *in vivo*, however the sequence signatures that drive it and the resultant structure signatures that define it are both unknown. To identify the signatures of CT folding and to estimate its abundance requires a set of sequences known to fold co-translationally. Unfortunately this set is not available for solved protein structures and so we use the Hydrophobic-Polar (HP) model (*Lau and Dill*, 1989). Within the HP model we can create a sequence set that folds co-translationally and, for comparison, a sequence set that does not. Here, we elucidate sequence and structure signatures that are related to the folding mechanism.

Previous studies that explored CT folding were based on hypothetical expectations and/or small data sets. We undertake the first exhaustive sequence classification based on folding ability. Our classification should enable any sequence and structure signatures of CT folding to be identified. Over 1.5 million sequences with unique global energy minimum (UGEM) are classified using our approach from Sec. 5.3.1. Thus, sequences are classified as

1. *Global-CT* - sequences reaching their UGEM conformation co-translationally.
2. *Kinetic-CT* - a final unique conformation is formed co-translationally but it is not the UGEM conformation.
3. *Non-CT* - no unique final CT conformation is attained, i.e. the sequence does not benefit from CT folding.

Comparison of our three sequence sets demonstrates that many presumed signatures of CT folding are too simplistic. For example, Global-CT conformations are not enriched in local nor previous contacts as postulated by *Alexandrov* (1993) & *Deane et al.* (2007). They have a more compact hydrophobic core, while the overall protein's core region shows more amino- than carboxy-terminal residues. Furthermore, we found the amino-terminus more centrally orientated than the C-terminus. The strongest signature of Global-CT folding is a restriction of the conformational space available to the amino-terminal region: a restriction that has been demonstrated in real protein structures (*Saunders and Deane*, 2009).

Analysing the SCOP database a general trend towards CT folding is observed. This is seen most strongly in the α/β class with 66% having a more centrally orientated amino- than carboxy-terminus and 71% having a more amino-terminal set of core residues.

In the following, we introduce the data sets and parameters used (Sec. 5.4.1). In Section 5.4.2 the features studied and the according algorithms and parameters used in the study are described, followed by the results for the HP model in Sec. 5.4.3. We investigate the presence of the identified signatures derived from the model in real protein structures from the SCOP database in Sec. 5.4.4.

5.4.1 Data and Setup

The "designing sequences" of *Irbäck and Sandelin* (2000) are used in the 2D-square lattice. "Designing sequences" are every HP sequence with a UGEM conformation (data available for sequence lengths $n \leq 25$). Our classification scheme from Sec. 5.3.1 grouped them into three folding sets: Global-CT, Kinetic-CT, and Non-CT. Additionally, all remaining HP sequences ($10 \leq n \leq 16$) were tested to see if they are Kinetic-CT sequences; that is whether they have a single conformation after $CT\ folding$ using the method from Sec. 5.3.1). All our 2D studies used a surmountable energy barrier of zero ($\Delta E = 0$) that prevented any chance of global folding.

In the 3D-cubic lattice, two non-exhaustive sets of $\sim 10,000$ and $17,000$) random sequences of length 27 and 36, resp., with a unique global fold were derived using HPoptdeg from the CPSP-tools package v2.4.2 (*Mann et al.*, 2008b). The method performs a degeneracy-decreasing local search in sequence space using the CPSP approach of *Backofen and Will* (2006) and is discussed in detail in Sec. 6.3.2. For our CT folding based sequence classification a surmountable energy barrier of 1 ($\Delta E = 1$) was used. Some of our experiments compare CT-folds to random structures, which are derived by applying $2n$ random *pull-moves* starting from the stretched chain.

Care must be taken when transferring measures designed on the HP model conformations to real protein structures. Proteins often have flexible, essentially unstructured, termini - the requirement for a single conformation in the HP model occludes these flexible tails from our analysis. Hence, we analyse protein structures from the start of the most N-terminal secondary structure element (SSE) to the end of the most C-terminal SSE. The program JOY (*Mizuguchi et al.*, 1998) is used to assign secondary structure. Only helix and strand covering amino acids in a run of three or more are considered to be SSEs.

A set of 10,311 domains from the SCOP database (*Murzin et al.*, 1995) (release 1.75) with a sequence identity cut-off of 40% was analysed. When comparing between SCOP classes only the 1969 α, 2174 β, 2652 α/β, and 2640 $\alpha+\beta$ domains were considered; other classes had less than 1000 occurrences.

5.4.2 Sequence and structure measures

Measures of sequence and structure properties are designed or adapted from the literature (*Huard et al.*, 2006; *Deane et al.*, 2007). Throughout, we consider backbone-only lattice protein structures in the HP model. Thus, a protein of length n is defined by its sequence $S = (S_1, \ldots, S_n)$ and its structure $P = (P_1, \ldots, P_n)$, where P_i denotes the position of the ith monomer. $\delta(P_i, P_j) = |P_i - P_j|$ denotes the Euclidean distance between the coordinates P_i and P_j and we define $h(S_i) = 1$ if S_i is hydrophobic and 0 otherwise.

In solved protein structures, S_1 is the most N-terminal and S_n the most C-terminal residue assigned to a run of at least three helix residues or three strand residues. The sequence record of PDB files is sometimes incomplete if not all residues are resolved in the X-ray structure - in these cases n is based on the actual residue number to incorporate chain breaks.

Sequence measures

- *Hydrophobicity* measures the percentage of residues in the sequence classified as hydrophobic ($= \frac{1}{n}\sum_i h(S_i)$). According to *Sandelin* (2004), hydrophobic residues are Ala, Cys, Ile, Leu, Met, Phe, Pro, Trp and Val.

- *Hydrophobicity by quartiles*, examines the Hydrophobicity of the sequence intervals given by $[1, \frac{n}{4}]$, $[\frac{n}{4}, \frac{n}{2}]$, $[\frac{n}{2}, \frac{3n}{4}]$, and $[\frac{3n}{4}, n]$.

- *Possible Hydrophobic Interactions* (*PHI*, Eq. 5.2) describes the relative possibility of each hydrophobic residue to make favourable (hydrophobic) interactions.

$$\mathit{PHI}(i) = \frac{\sum_{j\in[1,n]}^{|i-j|>d} I(S_i, S_j)}{n - 2d - 1} \tag{5.2}$$

where d defines the minimal distance in sequence considered for contacts and is set to 3 throughout. To evaluate interactions we use $I(S_i, S_j) = -1$ if both residues are hydrophobic ($h(S_i) = h(S_j) = 1$); otherwise $I(S_i, S_j) = +1$. Since the used rectangular lattices suffer the parity problem[1] (*Mann et al.*, 2008b), we halve the normalisation of the *PHI* score in lattice models using $(n - 2d - 1)/2$. For analysis, the *PHI* score is averaged for each position over all sequences per sequence set (Non-CT, Global-CT, Kinetic-CT).

- *Neutral nets* are network where: each node represents a protein sequence; all sequences (nodes) within the network share a common UGEM conformation and edges connect sequences that differ by a single point mutation. The most connected node, the *hub-node*, has been described as the most protein-like sequence because it is more robust to mutation (*Wroe et al.*, 2005). Neutral nets were built for the 2D HP sequences of $n = 25$. Only networks for structures that are the UGEM conformation for ≥ 50 sequences (including at least one Global-CT sequence) were considered. It is possible that the neutral network of a UGEM conformation contains more than one distinct connected component (*Wroe et al.*, 2005); in which case each component is analysed separately.

Structure measures

- The *Mean Central Residue (MCR, Eq. 5.3)* calculates the sequence position where the according residue/monomer position is closest to the protein's center Z, Eq. 5.5. This is similar to the measure *pMIN* from (*Deane et al.*, 2007).

$$MCR = \frac{\sum_{i=1}^{\eta} core(i) \cdot W(i)}{\sum_{i=1}^{\eta} n \cdot W(i)} \tag{5.3}$$

$$W(i) = \frac{1}{\delta(P_{core(i)}, Z)} \tag{5.4}$$

$$Z = \frac{1}{n} \sum_i P_i \tag{5.5}$$

Equation 5.3 utilises $core(k)$ to access the sequence index of the k-th closest residue P to Z according to $\delta(P, Z)$ (i.e. $\delta(P_{core(k)}, Z) \leq \delta(P_{core(k+1)}, Z)$). Through normalisation *MCR* maps to the interval $[0, 1]$; thus a more N-terminal set of η core residues has a score less than 0.5 while a greater value indicates a more central C-terminus. We used $\eta = 4$ and 8 in 2D and 3D respectively.

- NC_{cen} (Eq. 5.6), the *relative distance of the N and C termini to the protein's centre Z (Eq. 5.5),* assesses terminal bias of Z from a structural rather than sequence perspective. A negative NC_{cen} reveals that the N-terminus P_1 is more centrally orientated then the C-terminus P_n.

$$NC_{cen} = log \frac{\delta(P_1, Z)}{\delta(P_n, Z)} \tag{5.6}$$

- The *moment of inertia (MoI, Eq. 5.7)* measures structural compactness as the average squared distance of any residue to the center Z (*Huard et al.*, 2006).

$$MoI = \frac{1}{n} \sum_i [\delta(P_i, Z)]^2 \tag{5.7}$$

[1] Neighbored nodes in rectangular lattices show different parity in coordinate sum resulting in two classes of nodes, i.e. with even or odd coordinate sum. Due to the connectivity along the structure, only monomers with different sequence index parity can form contacts, i.e. even to odd and vice versa.

The *hydrophobic MoI* can be calculated when considering hydrophobic residues only; in this case n is the number of hydrophobic residues and Z is their center.

5.4.3 Results

Using our classification method from Sec. 5.3.1, we have classified HP sequences into three sets: Global-CT, Kinetic-CT, and Non-CT folders. The $765,147$ sequences of length 25 with a UGEM conformation in the 2D-square lattice (*Irbäck and Sandelin*, 2000) are separated into $17,085$ Global-CT, $74,502$ Kinetic-CT, and $673,560$ Non-CT sequences. In general, between 10 % and 20 % of our longer sequence sets seem to fold co-translationally. The percentage is higher for shorter sequences (43% at length 13) but this might be a bias from the relatively small structure spaces for these lengths. It suggests that a significant number of proteins may use co-translation to attain their biological conformation. The classification method from Sec. 5.3.1 is a very restrictive implementation of CT folding and many more sequences may use CT-folds as a springboard for finding their native conformation as we have seen from our classification evaluation in Sec. 5.3.2.

Global-CT sequences behave differently to the other sets when our measures of sequence space are considered. In our examination of neutral nets we find that Global-CT sequences are often the most robust to mutation. Hub nodes are enriched in Global-CT sequences; at length 25, Global-CT sequences make up just 12.8% of neutral net sequence space and account for 41.9% of hub sequences.

Structural compactness (*MoI*) did not segregate our sequence sets. However, examining the *hydrophobic MoI* did (see Fig. 5.3). In contrast to the theorised properties, Global-CT sequences had, on average, the most compact hydrophobic cores. Kinetic-CT sequences, in accordance with the theory, had the least compact cores; but the histogram distributions do overlap. It would appear that by selecting UGEM conformations that can be found co-translationally (from Global-CT sequences) we have also selected those conformations with the most compact hydrophobic cores. It may be that a compact core is related to the robustness to mutation exhibited by Global-CT sequences.

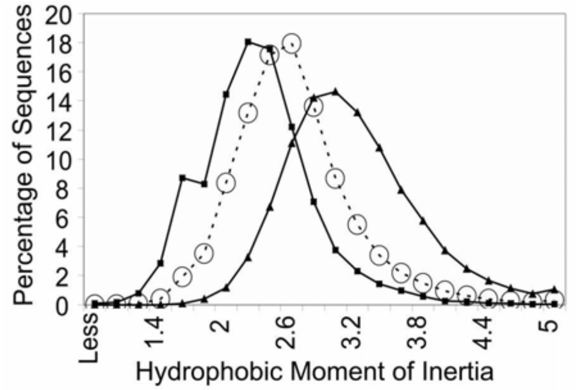


Figure 5.3: The frequency distribution of the *hydrophobic MoI* for Global-CT (filled squares), Non-CT (circles), and Kinetic-CT folders (filled triangles) of length 25 in the 2D-square lattice.

Undertaking a closer examination of core properties through the mean central residue (*MCR*) and the relative centrality of the N- and C-termini (NC_{cen}) we found, as expected, that Non-CT sequences show no overall bias. Our other sets do: 73% of Kinetic-CT and 75% of Global-CT conformations have a more centrally orientated N- than C-terminus at length 25. When considering the MCR, we found that 97% of Kinetic-CT and 93% of Global-CT sequences have an N-terminal core (length 25).

It was theorised that CT folding would produce a more compact N- than C-terminal region. We calculate the *MoI* of the extreme eight terminal residues and then compute $\log \frac{MoI(N)}{MoI(C)}$ where a negative result indicates a more compact N- than C-terminal region. Overall a negative score is observed for both Kinetic-CT and Global-CT sets. Non-CT folders have, on average, equally compact N- and C-termini.

Global-CT sequences also have a significantly lower percentage hydrophobicity (χ^2 test) and exhibit a general decrease in hydrophobicity moving from the N- to C-terminus as shown in Fig. 5.4.

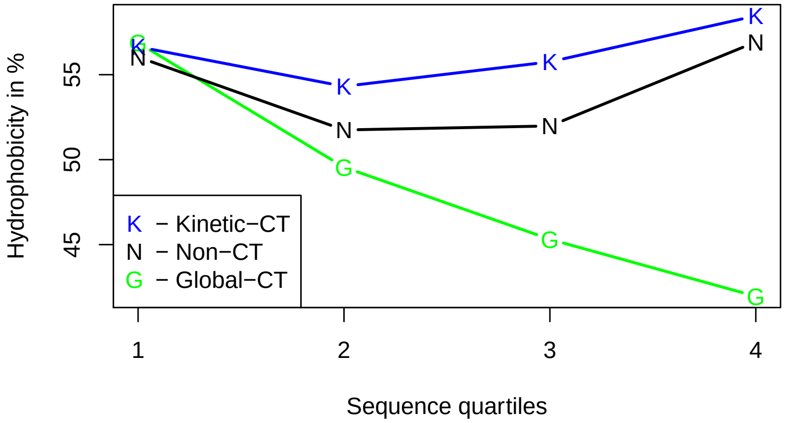


Figure 5.4: *Hydrophobicity by quartiles* at length 25 in the 2D-square lattice for Global-CT, Kinetic-CT and Non-CT folders.

Of all our tested measures, the possible hydrophobic interactions (*PHI*) score most clearly segregates Global-CT sequences from our other sequence sets. A positive *PHI* score for a position means that there are more unfavourable than favourable interactions possible at that position. Through our *PHI* score we demonstrate that Global-CT sequences are characterised by a low number of possible favourable (hydrophobic) interactions in N-terminal regions (positive *PHI*, see Figure 5.5). We suggest that specific contacts form in the N-terminal region that restrict and guide the subsequent folding process. The restriction on conformation space is unique to Global-CT sequences.

5.4.4 Lifting Model to Reality

Real protein structures have a modular design composed of domains, where each domain is assumed to be able to fold independently. Therefore, in order to study the possible signatures of CT folding within real protein structures the SCOP domain database is used.

As mentioned, care must be taken when expanding measures from the HP model to real protein domains and, as such, unstructured termini are occluded from our analysis. Additionally, it is unknown to which of our folding sets (Global-CT, Kinetic-CT, or Non-CT) each domain would belong. Hence, it is only possible to see if the data set as a whole is biased towards signatures of CT folding. We compare our measures between different SCOP classes as they have been suggested to vary in their propensity for CT folding (*Taylor*, 2006; *Deane et al.*, 2007). We have previously

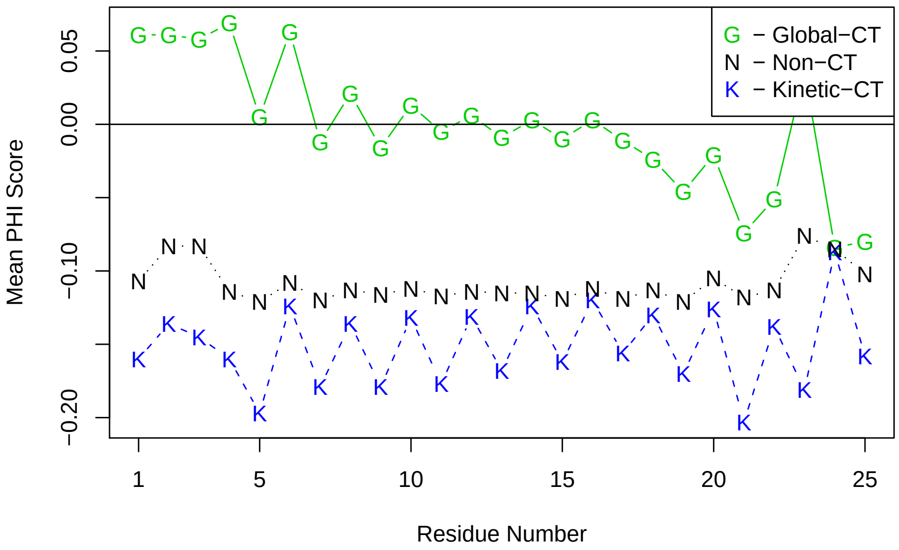


Figure 5.5: Possible Hydrophobic Interaction (*PHI*) score data for length 25 sequences in the 2D-square lattice. Global-CT sequences (G) have positive scores in the N-terminal region indicating that only a few hydrophobic interactions are possible, i.e. the majority of an H residue's potential contacts are with P residues and thus not favoured. For Kinetic-CT (K) and Non-CT (N) sequences' scores are always negative. Lines are a guide to the eye.

shown that, in contrast to *Laio and Micheletti* (2006), the N-terminal region of a SCOP domain is, on average, more compact than the C-terminal region (*Saunders and Deane*, 2009). In this manuscript we link this observation to CT folding.

SCOP domains in general exhibit a bias towards the signatures we ascribe to CT folding: 56% having an $MCR < 0.5$ and 58% having an $NC_{cen} < 0$. As expected, the bias varies between SCOP domain classes: α ($[MCR < 0.5] = 51\%$, $[NC_{cen} < 0] = 50\%$), β (47%, 60%), α/β (71%, 66%), and $\alpha{+}\beta$ (54%, 57%) (Fig. 5.6). The α class alone shows no bias in these tests. All other sets have a significant bias towards CT folding under our NC_{cen} measure; most strongly observed in the α/β class.

5.4.5 Discussion

Undertaking an exhaustive model study of co-translational (CT) folding, we have classified over 1.5 million HP sequences into three sequence sets (Global-CT, Kinetic-CT, and Non-CT) based on their CT folding properties. Global-CT sequences are optimised to find their unique global energy minimum (UGEM) conformation via a path of directed growth starting from their N-terminus (CT folding). Kinetic-CT sequences fold co-translationally to a unique final conformation but this conformation is not the unique global energy minimum. Non-CT sequences have a UGEM conformation but cannot attain this nor a unique final conformation co-translationally. A 1st-order Markov-chain simulation demonstrated the different folding behaviour of our sets and suggested that CT folding benefited all sequence sets in attaining their UGEM conformation. As expected, Global-CT sequences profited most while Kinetic-CT were significantly better folding into their kinetic fold identified by the classification procedure than into their UGEM structure.

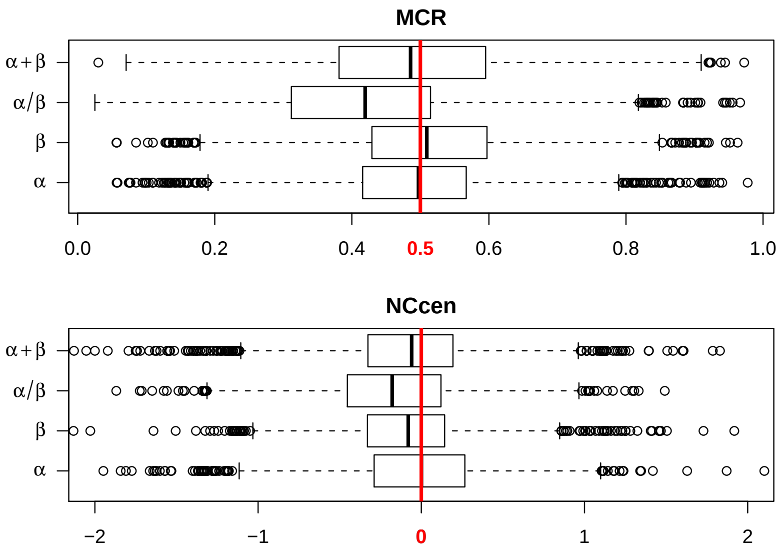


Figure 5.6: Distribution of the Mean Central Residue (MCR) and the relative distance of the N- and C-termini to the proein's center (NC_{cen}) for each SCOP class. Only the α class shows no bias according to the measures. All other classes, and at most the α/β class, show a trend towards CT folding behavior, i.e. $MCR < 0.5$ and $NC_{cen} < 0$.

Our three sequence sets allow the first test of the CT properties theorised in the literature (*Alexandrov*, 1993; *Laio and Micheletti*, 2006; *Deane et al.*, 2007). In general, signatures of CT folding are more subtle than theorised. CT conformations are not generally enriched in local nor previous contacts, nor are they less compact. We do find that CT conformations have a more N-terminal core; a more centrally orientated N- than C-terminus; and a more compact N- than C-terminal region. The real protein structures tested are biased towards these signatures of CT folding. As *Deane et al.* (2007), we find that SCOP classes differ in their propensity for CT folding properties and we highlight the α/β class as a strong candidate for CT folding.

Global-CT sequences dominate the hubs of large neutral nets and are thus, on average, more robust to mutation; a result consistent with that of *Wang and Klimov* (2007). *Xia and Levitt* (2004b) studied the evolution of folding rates and protein stability and demonstrated that there is a funnel-like organisation of sequence-space towards these hubs. *Govindarajan and Goldstein* (1998) suggested that CT folding may be the standard and that sequences evolve such that the structure found co-translationally becomes the UGEM conformation. In reference to this we find that some Kinetic-CT conformations are UGEM conformations for other sequences.

The most significant signature of CT folding we identified is a sequence-mediated restriction in N-terminal structure space. The N-terminus of Global-CT sequences can make relatively few favourable contacts and we suggest that the formation of these favourable bonds directs the rest of the folding pathway towards the UGEM structure. In this way a directed folding path is created as suggested by *Karplus* (1997). There is experimental evidence that local sequence effects on structure are stronger at the N- than C-terminus. Native N-terminal structures have recently been observed in otherwise denatured protein (*Chugha and Oas*, 2007). The restriction in structure space and stronger local sequence signals at the N-terminus should make prediction of structure in this region more accurate. Indeed increased prediction accuracy at the N-terminus was identified in secondary structure prediction by *Holley and Karplus* (1989) and has recently been revisited by *Saunders and Deane* (2009). Unfortunately our *PHI* score is currently not directly applicable to real protein data.

Overall, our results suggest that the abundance and importance of co-translational folding *in vivo* is currently underestimated.

5.5 Classification of Proteinlike Sequences in HP-models

Publication

Classifying protein-like sequences in arbitrary lattice protein models using LatPack
M. Mann, *D. Maticzka, R. Saunders, and R. Backofen*
in *HFSP Journal, Special Issue on Protein Folding: Experimental and Theoretical Approaches*, vol. 2(6), pp. 396, 2008
doi:10.2976/1.3027681, PMID: 19436498, PMCID: PMC2645588

In the last section, we presented a study to tackle one of the major biological questions: *"What are the features of proteins that enable unerring folding into their functional native structures rather than just producing random coils?"* To address such questions, comparative studies of protein sequence and structure space are necessary to identify underlying properties. Due to extreme computational complexity and limited knowledge of aspects governing protein folding it is currently not feasible to investigate the folding process of real proteins via full simulations nor to calculate their native structure directly. Thus abstract protein models as lattice proteons studied within this thesis have been defined to focus on and elucidate certain features of proteins and protein folding.

By reducing complexity, protein models are computationally accessible but induce a major problem: *One has to identify protein-like sequences!* Real protein sequences are usually not applicable or transferable due to model restrictions in sequence/structure space or simplified energy functions. The models' abstractions usually disrupt a reasonable one-to-one mapping of real and model sequences. Thus a biological protein sequence is not guaranteed to show protein-like (*in vivo*) behavior when ported into the model.

Therefore, a *model-specific classification/definition of protein-like sequences* has to be calculated for each protein model! Identified protein-like sequences must posses a (unique) stable native structure and, more importantly, be able to fold to this structure within a short (biologically relevant) time interval. Thus thermodynamic and kinetic properties have to be used. Without such a data set the study of the initially formulated question is inhibited by the unvalidated data underlying it. Here we introduce such a classification scheme of protein-likeness, essential for computationally accessible, biologically relevant models!

Our procedure is tailored to be used for lattice protein models as introduced in Sec. 1.2, independently from the used energy function or the underlying lattice. It is directly capable of work on any protein model due to the general applicability of the used LatPack tools (see methods or *Mann et al.* (2008a)). Quintessentially, we use a three-step classification system. First, thermodynamic features are checked in accordance with (*Crippen and Chhajer*, 2002) to ensure that a stable native structure can be found. Next, these sequences are tested to see if they can adopt their native structure in a short time interval. Thus a *good*/*bad* folder classification is achieved as done by (*Jacob and Unger*, 2007). Only *good* folders are considered in the final step where co-translational folding abilities are investigated. As we have shown in recent sections, co-translational folding is assumed to restrict the accessible parts of the energy landscape during folding and hence to guide the process to the native structure (*Levinthal*, 1968; *Govindarajan and Goldstein*, 1998). We therefore check presence of co-translational folding pathways with low energy barriers within the sequences' energy landscapes (*Huard et al.*, 2006). This final step accounts for the sequential assembly of proteins and therefore the occurrence of co-translational folding during elongation or membrane transports (*Fedorov and Baldwin*, 1997; *Kolb et al.*, 2000; *Deane et al.*, 2007).

The resulting *protein-like* sequences can thus be used to address the initial question. They can serve as *a reasonable data set* to investigate the features that enable their protein-like behavior via the study of their differences to random sequences.

Our approach is exemplified in the HP-model within a 3D-cubic lattice. We use our LatPack-tools package (*Mann et al.*, 2008a); a collection of programs and approaches to enable folding studies in the field of lattice proteins with arbitrary energy functions. The package is tailored to be as flexible as possible while ensuring high performance, essential for the computationally demanding tasks. So it is possible to perform the necessary kinetic folding simulations (LatFold) as well as sequential/co-translational folding studies (LatVec). The tools are described in more detail in the following section. Based on our classification we provide a large set of *protein-like*, *good* and *bad* folding sequences for the 3D-cubic HP-model. The data set is freely available, see Sec. 5.5.2.

In addition to the applicability of this data set to address relevant biological questions it can serve as the first well defined benchmark sequence set for folding algorithms (*Steinhöfel et al.*, 2007; *Dotu et al.*, 2008; *Ullah et al.*, 2009; **?**). So far new methods have usually been tested on random sequences that, with high-probability, will not show protein-like behavior. Since the approach and the used LatPack tools are applicable to arbitrary lattice protein models it opens the selection of such *data sets for any lattice protein model!*

5.5.1 Methods

As introduced above, we apply a three-step protocol to identify protein-like sequences. Such sequences have to have all of the following properties:

Thermodynamic stability: a unique global energy minimum (UGEM) structure has to exist that is most stable in thermodynamic equilibrium.

Fast folding kinetics: the sequence has to be able to attain the UGEM structure in a short time interval which implies a smooth, funnel-like energy landscape without kinetic traps.

Folding mechanism robustness: the UGEM structure has to be accessible via co-translational folding without high energy barriers, i.e. it has to be a Global-CT sequence according to Sec. 5.3.

In the following we detail the methods used to test the given properties. The protocol is exemplified for the widely studied backbone-only HP model in the 3D-cubic lattice.

Thermodynamic stability analysis in HP-models via HPdeg

As already discussed in the introduction, it is an essential feature of a protein-like sequence to have a thermodynamically stable native conformation. This results in the common assumption in simplified protein models that the structure of minimal energy corresponds to the native fold (*Crippen and Chhajer*, 2002; *Jacob and Unger*, 2007).

Unfortunately, the simple energy function in the HP-model (*Lau and Dill*, 1989) in tandem with the discretization of structure space (due to the lattice) induces a high degeneracy of the model (see Sec. 3.3.1). Thus, a high number of sequences have thousands or millions of structures with minimal energy. In order to allow for a stable native structure, such sequences cannot be considered as protein-like. Therefore, we are interested in sequences with a very low degeneracy or, even better, non-degenerated sequences. In earlier research it was felt that such non-degenerated sequences with a unique minimal energy structure would not exist (*Shakhnovich*, 1996). Due to the high computational complexity required to calculate even a single optimal structure (NP-complete) (*Berger and Leighton*, 1998) it was not thought possible to determine efficiently the degeneracy of a sequence, i.e. all optimal structures. Using the Constraint-based Protein Structure Prediction (CPSP) approach by *Backofen and Will* (2006) introduced in Sec. 3.2.1, we have shown in (*Mann et al.*, 2008b) that such structures exist and can be detected with very low time consumption.

Here, we use the CPSP-implementation HPDEG to test if a sequence can adopt only a single minimal energy structure, i.e. has a degeneracy of 1. Using HPSTRUCT we can directly predict this UGEM structure or (if the degeneracy is larger) all energy minimal structures. Thus, given a set of sequences, we can identify all *non-degenerated folding sequences* within the set and use them for further investigations.

The problem of a high average degeneracy is common in lattice protein models. It results mainly from the discretization of sequence and structure space and the simple energy function applied. Thus an approach for the calculation of a sequence's degeneracy would be needed for each model, as we use the CPSP approach for the HP-model. Currently, only for the HPNX-model (*Renner and Bornberg-Bauer*, 1997; *Wolfinger et al.*, 2006) does such an approach exist - an extension of the CPSP approach (*Backofen and Will*, 1998, 2006).

In some cases, the restriction to non-degenerated sequences might be too severe and also sequences with a low degeneracy are of interest. The CPSP-tools, used for the degeneracy classification in the HP-model, support the identification of these sequences too (*Mann et al.*, 2008b). The number of sequences grows exponentially with rising degeneracy in the HP-model. Nevertheless, the modularity of our classification protocol is well suited to incorporate such customisations.

Fast folding kinetics investigation via LatFold

As done in Sec. 5.2, we use the program LATFOLD for global folding simulations of lattice proteins. Within LATFOLD, the folding path is emulated via a 1st-order Monte-Carlo simulation using a Metropolis criterion as done in (*Jacob and Unger*, 2007; *Thachuk et al.*, 2007). It is therefore an iterative procedure that at each step takes the current structure and, using a move set, identifies a neighbored (very similar) structure in the energy landscape. The temperature-depending Metropolis criterion is used to determine if the neighboring structure is accepted or rejected based on the energy difference of both structures (Eq. 1.33). The method is discussed in detail in Sec. 1.4.2.1.

The program is applicable to lattice protein models with arbitrary contact or distance based energy functions and is consequently very general. Furthermore the energy function can be chosen independently from the lattice used. We are currently supporting the unrestricted simple 2D-square, 3D-cubic and the highly complex 3D-face-centered-cubic lattice. An extension of LatFold to other lattices is easily possible due to its modular implementation.

To generate neighboring structures within LatFold, two different ergodic so called *move sets* can be chosen. Move sets are generic definitions of rules to apply (small) structural changes within a given protein conformation to generate structures neighbored in the energy landscape. Thus an iterative application of such *moves* models structural changes over time, i.e. folding. The ergodicity ensures that all structures can be transformed into each other via a sequence of moves. *Pivot-moves* introduced by *Madras and Sokal* (1988) yield relatively strong structural changes (*Wolfinger et al.*, 2006) by rotating huge parts of the structure. In contrast, the application of *pull-moves* defined by *Lesh et al.* (2003) results in more local changes of the structure. It has been shown that *pull-moves* are able to reproduce realistic folding times (*Steinhöfel et al.*, 2007) and are therefore well suited for our folding based classification similar to (*Jacob and Unger*, 2007). For triangular latties, as the FCC, we support the according pull move adaption by *Böckenhauer et al.* (2008). To our knowledge there are no other ergodic move sets for lattice protein models. The non-ergodic local moves (*Madras and Sokal*, 1987) are not used due to the partitioning of the accessible energy landscape into independent ergodicity classes. Their number is growing exponentially while each ergodicity class gets exponentially small.

The folding kinetics of a protein sequence S with a unique global energy minimum structure P_{UGEM} are studied as follows. Given a move set defining the neighborhood relation $\mathcal{N}$ and a relative folding temperature kT, we perform a large number of folding simulations using the LatFold program. Each simulation is stopped if either the ground state P_{UGEM} is reached or a given number of simulation steps is exceeded. It is only considered a successful run if the structure P_{UGEM} was reached within the limited simulation time. The rate of successful runs over all performed folding simulations enables an identification of *fast folding sequences*, i.e. sequences that have a high success rate for a statistically sound number of simulations.

Since the used Metropolis Monte-Carlo simulation is temperature dependent, we have to identify the correct folding temperature kT to be applied for the described classification. To this end, we apply a temperature screening introduced in the following section.

Determination of the optimal folding temperature

The Metropolis criterion from Eq. 1.33 uses the Boltzmann weight $e^{-\frac{E(P_{\text{to}})-E(P_{\text{from}})}{k_B T}}$ of the energy difference of two structures $P_{\text{from}}, P_{\text{to}}$ to determine the probability of the folding transition $P_{\text{from}} \rightarrow P_{\text{to}}$. Due to the coarse grained energy function, we can neither assume that folding happens at room temperature (T) nor can we apply the Boltzmann factor k_B since lattice models' energy functions are typically arbitrarily chosen and do not represent measured physical values! Furthermore, the optimal folding temperature T_f, where the native structure P_{UGEM} of a protein S is adopted best and is stable too, is unknown and has to be determined for each protein model independently.

It is sufficient to determine the product kT_f instead of T_f and k independently. This is achieved by a screening with folding simulations (using LatFold) over different values of kT for a non-redundant set of non-degenerated sequences, because we are only interested in their folding behavior and not in random sequence folding. We define kT_f as the value where the folding simulations spend most of the time in the native state.

We expect that folding simulations with low kT are usually early trapped in local minima (non-native structures) of the energy landscape and thus do not or rarely reach the unique global energy minimum. For high kT, a randomized traversal of the landscape is expected, resulting in a few native fold hits and a high variety in the adopted energies. At kT_f the simulation should hit the native structure at high rate and stay there for long periods.

Figure 5.7 shows representative screening simulation trajectories (energy runs) that exemplify the expected behavior for different kT values in the HP-model. The simulations are performed using *pull-moves* in the 3D-cubic lattice. For very low kT the simulation is immediately trapped while for high values a random behavior is observed. Only in the 3rd plot at kT_f, the energy of the *single* native structure is reached, kept and recovered (if left) over long time series.

Folding mechanism robustness check via LatVec

As discussed in the introduction of this chapter (Sec. 5.1), proteins are assembled in a sequential manner at the ribosome. Thus it is very likely and in some instances has been experimentally verified that the protein begins to fold before it is released from the ribosome (*Frydman et al.*, 1994; *Nicola et al.*, 1999; *Kolb et al.*, 2000; *Kolb*, 2001). As we have shown in Sec. 5.4, co-translational folding seems to be a powerful and prominent folding mechanism and sequences that are able to use it should be more robust in their folding behaviour. So far, our classification protocol does not consider this co-translational scenario and assumes global folding of the whole protein as occurring e.g. after unfolding of the structure due to heat shock or other environmental changes.

To integrate a protein's co-translational folding ability into our classification we revisit our set of *fast* folders from the last section. We want to further partition this set based on the ability of each sequence to fold co-translationally, assuming that this feature describes an additional fundamental property of proteins (see Sec. 5.4). Since we have focused on sequences that fold into their unique global energy minimum (UGEM) structure P_{UGEM}, we want to identify Global-CT-folders among our fast global folders using the approach from Sec. 5.3. Global folders are sequences that show no energy barriers along co-translation folding pathways to P_{UGEM} that exceed a given low allowed refolding energy threshold. Thus, co-translationalq folding directly guides the folding process towards the UGEM structure resulting in robust folding.

Given the set of fast folders from the last section, we identify all *Global-CT sequences* within this set. These sequences are robust to the applied folding mechanism since they show a good global folding behaviour (fast folding) and show no high energy barriers along co-translational folding pathways into the UGEM structure.

Proteinlike Sequences

Combining the steps from above, we define a sequence *protein-like* if it shows the following properties (exemplified in the HP-model):

1. It is *non-degenerated*, i.e. HPdeg confirms a degeneracy of 1. The unique ground state P_{UGEM} is calculated using HPstruct.

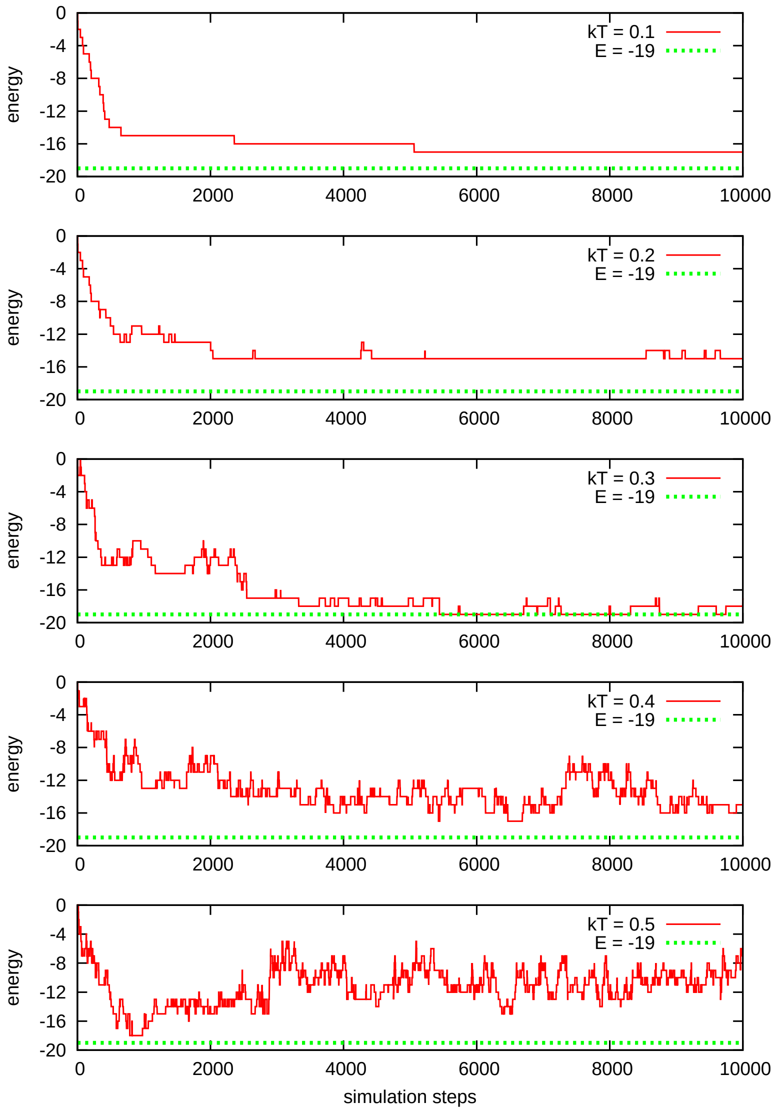


Figure 5.7: Folding simulations for a kT series to identify the ideal relative folding temperature kT_f (3rd plot). The dotted green line marks the energy of the unique native structure, i.e. if it is reached the native structure is adopted.

2. The sequence is a *good/fast folder*, i.e. it reaches very often its ground state structure P_{UGEM} within short global folding simulations using LatFold (given the optimal folding temperature kT_f, which was found to be 0.3 within this model).

3. Using LatVec, the sequence was classified *Global-CT*, i.e. along co-translational folding pathways it can reach P_{UGEM} without passing large energy barriers ($\Delta E = 2$).

In the following, we will exemplify the whole classification protocol with the aim of producing a large benchmark set to be available for future studies.

5.5.2 Results

In the following we will demonstrate our strategy to classify protein-like sequences in simplified lattice protein models based on folding properties. We utilize the HP-model, but the strategy is directly applicable to arbitrary lattice protein models.

The unrestricted 3D-cubic lattice was chosen due to its prevalence in previous protein studies (*Jacob et al.*, 2007; *Steinhöfel et al.*, 2007; *Wolfinger et al.*, 2006; *Jacob and Unger*, 2007; *Thachuk et al.*, 2007) and the abundance of reasonable sequence sets. Often the used benchmark sets consider degeneracy only and thus (with the exception of (*Jacob and Unger*, 2007)) do not reflect a reasonable protein-likeness definition based on kinetic properties. Furthermore, they usually consist of a few sequences only. Here we implement a generic, transparent and reproducible definition with the aim of producing a large benchmark set for use in future studies.

The classification is mainly achieved using folding simulations. For global folding, where the whole fold space is explored, we use the *pull-move* set as introduced by (*Lesh et al.*, 2003). This set is often used (*Thachuk et al.*, 2007; *Ullah et al.*, 2009) and has been shown to yield realistic folding times (*Steinhöfel et al.*, 2007). We address the problem of correct folding temperatures essential for reasonable Monte-Carlo simulations (see methods). The outcome of our procedure is a data set consisting of *protein-like* sequences, *good (fast)* and *bad* folders that is freely accessible at

`http://www.bioinf.uni-freiburg.de/Data/`

Non-degenerated Sequences

In the first classification step of our approach we search for *non-degenerated* sequences. Thus to classify a sequence as protein-like we assume, as the minimal requirement, that the sequence enables only a unique native structure P_{UGEM}. Using the CPSP-tools (*Mann et al.*, 2008b) we observed that only about 0.01-0.02 percent of all sequences fulfill this property in the unrestricted 3D-cubic HP-model (see Sec. 3.3). Thus only a small fraction of sequences are considered in the next, kinetic-based, classification step.

For illustration, we derived a random non-exhaustive set of 10,500 *non-degenerated* HP-sequences of length 27 and the according ground state structure P_{UGEM}. This sequence set will be used in the following to exemplify the whole classification approach.

Fast Folding Sequences

To prepare the folding kinetics analysis of the sequences, we have performed a kT screening for a subset of the sequence set. To this end, a non-redundant set of 50 sequences were selected at random from the pool of 10,500 non-degenerated sequences from the first classification step. For each sequence at every kT-value screened, 1000 folding simulations with 10,000 steps were done. For each kT value and sequence we calculate the mean number of steps where the P_{UGEM} structure was adopted. This way we could determine the individual optimal folding temperature kT_f^i for each screened sequence. To derive a general folding temperature kT_f for all sequences, we averaged over all gained kT_f^i values.

The resulting kT_f for the non-degenerated HP-sequences of length 27 in the unrestricted 3D-cubic lattice using *pull-moves* is $kT_f \sim 0.3$ (in detail 0.285). We observe a very low variance of 0.006. Thus, we did not increase our low sample size of 50 sequences for the kT screen. Independent tests revealed the same kT_f characteristics for the resulting classified groups (data not shown). Thus the kT_f choice seems to be invariant to the specific sequence set used in this model. In general a higher sample size should be used if the kT values show a higher variance.

Our determined kT_f is close to the folding temperature ($kT = 0.5$) for sequences of length 25 in the 2D-square lattice as used by *Jacob and Unger* (2007); however, it is unclear (in contrast to our value) how Jacob and Unger decided on this value.

Given the optimal kT_f value, we are now in the position to classify lattice protein sequences based on kinetic properties. The goal of the second classification step is to separate our *non-degenerated* sequences into the two sets *good and bad folders*, depending on their kinetic properties. Good folders are assumed to be the more protein-like sequences due to the ability to fold into their native structure very fast. On the opposite the bad folders represent random protein sequences that are able to form a random coil but no stable functional native structure (*Mazzoni and Casetti*, 2006). Such a classification enables new studies to investigate the common properties of good vs. bad folders; perhaps facilitating identification of the properties that allow for folding into a unique native state (*Jacob and Unger*, 2007). This property is often assumed to correlate with a feature of the energy landscape, the so called *folding funnel*. For good folders such a funnel is assumed to cover large parts of the landscape and drives the folding process downwards to the native fold (*Wolynes et al.*, 1995; *Klemm et al.*, 2008). If no folding funnel exists, as expected for random sequences, folding is assumed to do a "blind" search in the protein's rugged energy landscape.

We are going to do a *good/bad* folder classification for the large non-redundant set of non-degenerated HP sequences of length 27 from the first classification step. For each sequence we perform a series of 1000 very short folding simulations with 4000 steps at the given value kT_f using LatFold to allow for reasonable statistics and a high parallelization of the computations. The choice of 4000 steps was based on preliminary tests (data not shown) and has to be adopted for each protein model and length. We stop a simulation early if the native structure is adopted. Therefore, we are able to measure how often a sequence is able to adopt its native conformation in a given short time interval. This is of importance to mimic the relatively short folding time of proteins observed *in vivo*. Once the native structure is reached we assume it is kept because we are simulating at the optimal folding temperature (kT_f used).

A histogram on the "success rates" of the sequences is given in Figure 5.8. For each sequence the number of successful runs that found the native structure out of the 1000 runs was determined (bins of the histogram). The label of each bin gives the lower bound on the interval the bin covers to allow for a logarithmic view. The observed numbers of hits lies in the range 0 to 125, thus the sequence with the highest success rate found its native structure in 12.5% of the short runs. These

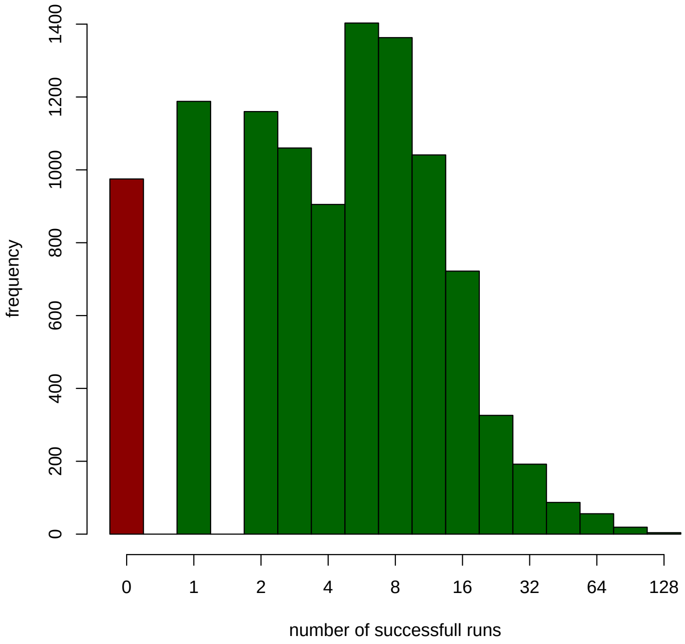


Figure 5.8: Histogram of the sequence numbers based on successful runs that end in the native structure out of 1000 runs with maximal 4000 simulation steps per sequence (green bars). The red bar on the left represents the number of sequences that did not found their native structure within the given simulation length.

are the best candidates for *good* folders. The number of sequences not able to fold into their native structures within the given simulation time is about 10%. Furthermore it becomes visible that a low number of hits is a common feature (about 70% show 1-9 successful runs). The wide range of hit-ratios allows for a setup of thresholds to classify sequences into good and bad folders. Based on the collected data on the 10,500 sequences we can set two thresholds. h_{bad} marks the maximal number of hits to mark a sequence still as a *bad* folder and h_{good} the minimal hit number for a sequence to be classified as a *good* folder.

For our data, we set $h_{\text{bad}} = 1$ and $h_{\text{good}} = 10$ to split the data set into quartiles. Based on this setting we get 3 classes of sequences: 2163 *bad* folder, 2447 *good* folder and 5890 "in-between" not classified *non-degenerated* sequences with a hit rate in the interval (h_{bad}, h_{good}) excluding the limits. This leaves a large set of good folders for the last classification step.

Global-CT Sequences

For all 2447 *good* folders, we run a sequential folding simulation using LatVec. To prevent sequences from becoming trapped in shallow, local energy minima we allow sequences to overcome small energy barriers in the co-translational folding pathway. We perform co-translational folding simulations allowing for a maximal surmountable energy barriers $\Delta E = 2$. For each sequence we check if and on which energy threshold the UGEM structure is reachable. If the UGEM structure is accessible under the given conditions it is classified a Global-CT folder, see Sec. 5.3 for further details. Thus, the protein benefits from the structural restrictions while it is assembled at the ribosome and its co-translational folding process is guided towards its native structure.

Proteinlike Sequences

Based on the last classification step, we end up with 605 *protein-like* sequences. Thus, these sequences are non-degenerated and show a unique global energy minimal (UGEM) structure; they are good folders and show no high energy barriers along co-translational folding pathways into their UGEM structure, i.e. are Global-CT sequences.

A first screen on sequence features did not reveal significant differences to one of the other sequence classes. Only the average rate to find the UGEM structure via short folding simulations is slightly higher for protein-like sequences (0.21%) compared to non-protein-like good folder (0.18%).

5.5.3 Discussion

The selection of protein-like sequences is an important problem in simplified protein models. The use of such sequences opens is essential to make studies on folding kinetics, sequence evolution and docking experiments reasonable. Currently, within abstract but computationally accessible lattice protein models often only thermodynamic criteria are considered for their selection. As we have seen from our results, such a selection is not sufficient when interested in sequences that show a protein-like folding behaviour. This becomes even worse if random sequences are used that are even less likely to show protein-like properties.

We introduce a three-step classification scheme that incorporates both the thermodynamic features usually applied as well as kinetic properties of sequences. At first, the thermodynamic stability of a sequence is checked, e.g. based on its degeneracy. Furthermore, the sequence has to be able to adopt its (unique) native structure in a short folding time (*good* folder behavior). Finally, we take into account the sequential assembly of proteins and evaluate their co-translational folding behaviour. Here each sequence is checked to see if its native structure can be adopted with co-translational folding if only *small* energy barriers are allowed to be overcome. Only if all three properties are present, the sequence is classified to be *protein-like*.

This classification scheme was applied to a non-exhaustive set of 10,500 random non-degenerated sequences of length 27 in the 3D-cubic HP-model. We end up with 4 sequence sets that are available online (see results):

- 605 *protein-like* sequences that fulfill all requirements, i.e. are non-degenerated good folders classified as Global-CT sequences,
- 1842 non-protein-like *good* folders that are no Global-CT sequences,
- 2163 *bad* folders that can not fold into their UGEM structure in short folding simulations, and
- 5890 unclassified *non-degenerated* sequences with "intermediate" folding behaviour.

This data set is the first classification based on thermodynamic *and* kinetic features that respects the sequential production of proteins as well. It can therefore serve as the basis for more profound studies in abstract models.

Though only demonstrated here for short sequence lengths in the simple 3D HP-model, the whole classification approach is applicable to arbitrary lattice protein models using a contact or even distance based energy function. The used tools LatFold and LatVec are able to perform the necessary folding simulations for any of these models. Thus the classification can be done for any sequence set and model of interest using our freely available LatPack tools.

CHAPTER VI

Neutral Evolution in HP-models

6.1 Introduction

Motoo Kimura formulated in the 60s and 70s the theory of neutral evolution (*Kimura*, 1968, 1983). *Kimura* (1991) states: “The neutral theory claims that the overwhelming majority of evolutionary changes at the molecular level are caused by random fixation (due to random sampling drift in finite populations) of selectively neutral (i.e., selectively equivalent) mutants under continued inputs of mutations. The theory also asserts that most of the genetic variability within species at the molecular level (such as protein and DNA polymorphism) are selectively neutral or very nearly neutral and that they are maintained in the species by the balance between mutational input and random extinction.”

Thus, most mutations during evolution are supposed to not influence the fitness of the organism, i.e. they are evolutionary neutral and not “visible” to natural selection proposed by Darwin. When focusing on molecules like proteins, fitness is correlated to the ability to perform a certain function. But such a function typically requires the stable and robust formation of a specific three-dimensional structure. According to the neutral theory, a protein is supposed to be more robust to mutations if its structure, i.e. function, can be encoded by a wide variety of different sequences. Thus, a single mutation is less likely to alter the structure which would (in most cases) reduce fitness (*Wroe et al.*, 2005). The theory is supported by the obvious existence of large sets of quite different protein sequences that are associated to the same function that is mediated by the formation of a similar protein fold, a so called motif.

A well known example is the zinc finger motif associated with DNA-binding. The ExPASy Prosite database (*Gasteiger E. and A.*, 2003) of the Swiss Institute of Bioinformatics (SIB) lists currently 158 sequences for the GATA family of transcription factors sharing the zinc finger motif shown in Fig. 6.1 (ID `PDOC00300`)[1]. The Figure also shows the sequence logo of the zinc finger encoding subsequences. From the 31 sequence positions only half is conserved in sequence, most of them involved in the binding of the zinc ion needed to stabilize the structure. The remaining sequence positions show a wide variety in the found amino acids as predicted by the neutral theory.

Given the neutral theory, i.e. that most mutations do not influence fitness, evolution is supposed to be a more or less random process of genetic drift. Within the latter, the frequencies of a given sequence in a (limited) population follows random sampling . *Wagner* (2008) postulates that evo-

[1] Accessed on 7th March 2011.

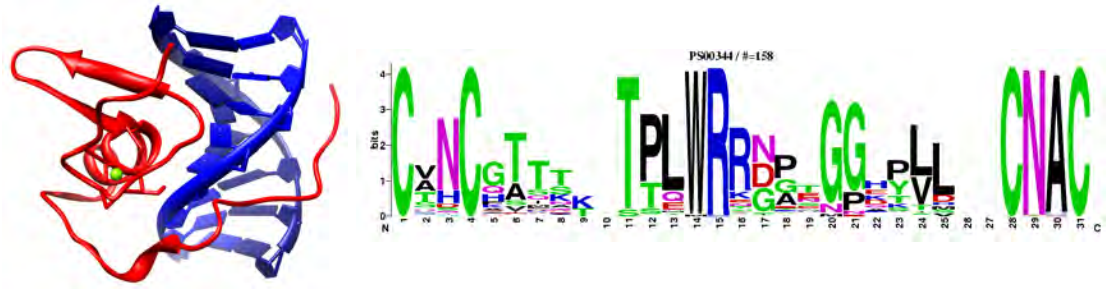


Figure 6.1: (left) The complexed DNA with the zinc containing DNA binding domain of the erythroid transcription factor GATA-1 (PDBID 1GAT). The DNA strands are given in blue, the protein subchain in red, and the zinc ion is colored green. (right) Sequence logo from the alignment of 158 sequences encoding the DNA binding domain from the left from the ExPASy Prosite database (ID PS00344). The amino acid letter size for a given sequence position corresponds to its dominance within the according alignment column.

lution can be seen as a stepwise combination of such random drift in sequence space via neutral mutations and the fixation of rare positive changes within the population via natural selection. A similar conclusion was drawn by *Tokuriki et al.* (2008).

On the other hand, it has been widely discussed that mutation rate μ and population size N influence the evolutionary effect of neutral mutations (*Kimura*, 1983; *van Nimwegen et al.*, 1999). In case of $\mu N \ll 1$ the random behaviour predicted by the neutral theory is observed. In contrast, a very high mutation rate or large population size ($\mu N \gg 1$) results in a selection for more mutation robust sequences, i.e. sequences that enable more neutral mutations (*Wagner*, 2008). This behaviour was connected to a certain topological characteristic in sequence space named the "superfunnel" (*Xia and Levitt*, 2004b; *Bornberg-Bauer*, 1997b; *Bornberg-Bauer and Chan*, 1999; *Wroe et al.*, 2005). The idea is that mutationally highly robust sequences are surrounded by less robust ones and so forth. Thus, if the situation demands for robust sequences ($\mu N \gg 1$) evolution is guided towards the central sequences (*Noirel and Simonson*, 2008).

A common approach to study superfunnels and neutral evolution is to investigate so called neutral networks (*Schuster et al.*, 1994; *Tacker et al.*, 1996; *Schuster*, 2008). A neutral network is an undirected, unrooted graph that represents the part of the sequence space accessible to neutral evolution for a given structure. To this end, each node represents a sequence that forms the structure as its unique energy minimal structure. If two sequences can be transformed into each other via mutation, e.g. by a single point mutation, they are connected via an edge in the neutral network. Thus, two sequences are connected if evolution can proceed from one to another without altering the function in accordance with the neutral theory (*Reidys*, 2009). Therefore, the more connected a sequence node, the more robust it is to maintain its function if mutated. *Wroe et al.* (2005) named these nodes "hub"-nodes or prototype sequence and they correspond to the central sinks of superfunnels discussed above.

Neutral networks have been shown to be not necessarily connected (*Reidys et al.*, 2006). Within a 2D-square HP-model confined to maximally compact, rectangular structures, *Noirel and Simonson* (2008) found that neutral networks usually contain one giant connected component and a number of very small ones. Nevertheless, neutral networks are relatively small compared to the vast size of the overall sequence space.

As already discussed, even if many mutations during evolution are neutral, there is still a high chance that a mutation will reduce a protein's fitness. This can either occur due to a destabilization of the functional fold (e.g. it is not the unique global energy minimum (UGEM) anymore) or via a reshaping of the energy landscape such that the folding process is disturbed (e.g. longer folding times) or ends in a non-functional structure. When considering the inverse folding problem, i.e. the search for a sequence that folds into a given structure, it becomes clear that one is looking for a "needle in a haystack". *Hart* (1997) has shown that the inverse folding problem is NP-hard in the 2D-square HP lattice models if the given structure has to be the UGEM structure for the sequence to be found. The results were later revisited by *Berman et al.* (2007) and extended to the 3D-cubic lattice.

Plan of the Chapter

In the following Section 6.2, we will introduce a new approach to solve the inverse folding problem. The new heuristic enables the study of neutral networks with multiple connected components in 3D-lattice models (presented in Sec. 6.3) without full sequence space enumeration. The latter was so far the only and standard procedure in the field (*Bornberg-Bauer*, 2002; *Koh et al.*, 2005). We show that neutral networks in these models show quite different characteristics compared to 2D-models discussed in literature. The networks are much smaller and less prominent in sequence space for 3D-models. This behaviour results from the increased structural flexibility within the model such that it becomes less likely that (a) a single UGEM structure can be formed and that (b) many sequences form the same UGEM structure, i.e. are within the same neutral network. To tackle these problems, we introduce a new perspective on structure space in form of H-folds in Sec. 6.4. Our view focuses on substructures formed by H-monomers only and their relative arrangement. This idea corresponds very well with our current understanding of real protein structures, where only the central structural elements are known to be stable and where the connecting loop regions are quite flexible.

6.2 Inverse Folding Heuristics

As already briefly introduced in Sec. 1.3.4, the *Inverse Folding* or *Sequence Design* problem is to determine a protein sequence for a given protein structure such that some given optimization criterion is met.

A common goal is to find a sequence that forms the given structure as its global energy minimum structure. To this end, *Yue and Dill* (1992) studied different sequence and structure features to enable a sequence prediction. Based on their studies they assumed that it is not possible to design HP sequences that form a given structure as its unique optimal structure (*Yue et al.*, 1995).

It was shown by *Hart* (1997) that the inverse folding problem (IFP) is a hard NP-complete problem. Later, *Berman et al.* (2004, 2007) revisited the problem and found the IFP only NP-complete in 3D-lattices, not in 2D-lattices. To tackle the IFP, several approximating approaches have been suggested, among them are (*Gupta et al.*, 2005; *Koh et al.*, 2005; *Khodabakhshi et al.*, 2009)

Here, we introduce a new approach named HPDESIGN that solves the inverse folding problem for backbone-only HP-models in 3D lattices. Given a structure P, HPDESIGN identifies sequences S that form P as their global optimum structure. Furthermore, the sequence's degeneracy is limited to ensure that P is the unique optimum. The latter property can be relaxed to allow for a limited degeneracy. To solve the problem, the procedure uses the CPSP approach introduced in Sec. 3.2.1 and the underlying H-core database. The heuristics follows an iterative *Generate-and-Test* scheme with two steps. First, using a structural filtering based on hydrophobic cores, a set of candidate sequences is derived. Afterwards, each sequence from this set is checked via the application of the CPSP approach if the structure P is among the optimal ones and if the degeneracy limit is maintained.

In the following, we give a detailed description of the HPDESIGN approach. The method is than evaluated in Sec. 6.3 for HP sequences of length 27 and 36 in the 3D-cubic lattice to study neutral networks.

6.2.1 Approach

HPDESIGN allows for the enumeration of sequences that form a given structure as their unique global energy minimum structure and thus solves the inverse folding problem (IFP). It is a Generate-And-Test method in two steps that are introduced in the following.

6.2.1.1 Step 1 : Candidate Set Generation

The number of possible sequences for a given structure P is exponential in its length n and the size of the sequence's alphabet. For the HP-Model it is therefore 2^n. To enable a Generate-And-Test approach we have to keep the number of generated sequences to test as small as possible. In HPDESIGN, this is done using H-cores introduced for the CPSP approach in Sec. 3.2.1. As sketched in Fig. 6.2, a sequence combined with a structure defines an H-core, i.e. the set of positions occupied by H-monomers. But the same holds in the reverse case, i.e. a structure and a fitting H-core determine a sequence.

This observation can be used to restrict the generated candidate set as follows. Let's recollect the targeted sequence criteria: the final sequences have to form the given structure P as a global energy minimal one. As we have already discussed in Sec. 3.2.1, in the HP-model optimal structures show a (nearly) optimal H-core. Thus, reversing the perspective as given above, if we can fit an optimal H-core into the structure P it is quite likely that the resulting sequence will have this structure as an optimal one. Therefore, we will use optimal and suboptimal H-cores to generate candidate sequences for a given structure.

The set of optimal H-cores of size k is given by $\mathcal{H}_k$, i.e. each H-core $h \in \mathcal{H}_k$ resembles a compact placement of k monomers in a given lattice L that show the maximal number of contacts between each other. The suboptimal H-cores with at least one less contact than $h \in \mathcal{H}_k$ are collected in the H-core set $\mathcal{H}'_k$. They form the first level of suboptimality. Thus, the next suboptimal H-cores of the second level of suboptimality are within $\mathcal{H}''_k$ and so forth.

Figures 6.3 and 6.4 depict the number of H-cores within $\mathcal{H}_k, \mathcal{H}'_k$ and $\mathcal{H}''_k$ for various sizes k in the 3D-cubic and 3D-FCC lattice, respectively. The number of optimal H-cores is usually quite small. Their number growths exponentially with increasing level of suboptimality. Nevertheless,

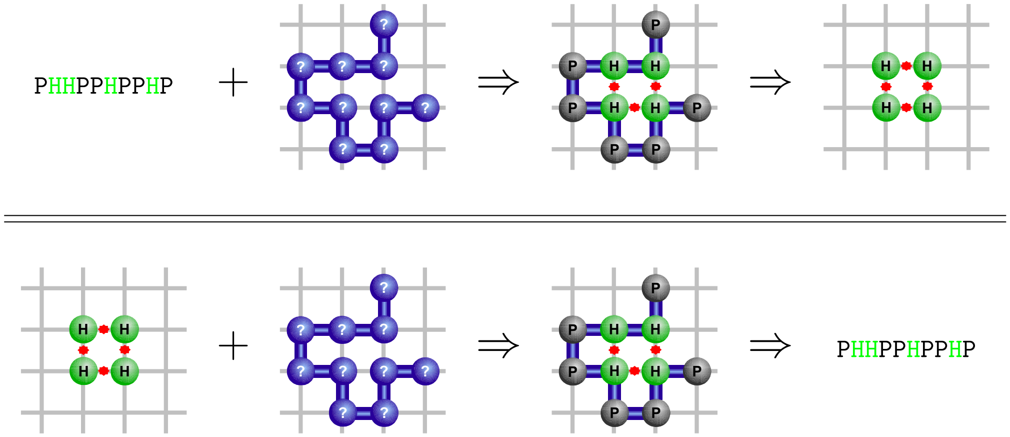


Figure 6.2: The relation between sequence, structure, and H-core of a structure in the HP-model. (top) The combination of sequence and structure defines an H-core. But the same holds vice versa for a positioned H-core within a structure (bottom).

when considering only a limited suboptimality their number is still much lower than the number of possible sequences.

The candidate sequence generation for a given structure P of length n in a lattice L can be decomposed based on the number of H-monomers in the targeted sequences. Thus, for each appropriate number of H-monomers k in the final sequences, we apply the following procedure:

We initialize the set of H-cores to process $\mathcal{H}$ with as many levels of suboptimal H-cores of size k as are of interest. Here, we restrict to the second level of suboptimality. Therefore, we set $\mathcal{H} = \mathcal{H}_k \cup \mathcal{H}'_k \cup \mathcal{H}''_k$. For each of these H-cores $h \in \mathcal{H}$, we map the core onto the target structure P to find all non-symmetric superpositionings of h and P. Thus, for every superpositioning we store the resulting candidate sequence S in $\mathcal{S}$.

The procedure is repeated for all H-core sizes of interest. Following our observations that non-degenerated sequences have on average an H-content of at least 40%, we set $k > 0.4{\cdot}n$. The sequences from this collected set of candidates $\mathcal{S}$ have a high chance to form P as an optimal structure and will be investigated in the following filtering step of HPdesign.

An example illustrating the candidate sequence generation is given in Fig. 6.5. Here, the H-core of Fig. 6.2 can be superpositioned in three ways on the given structure and yields three different candidate sequences.

6.2.1.2 Step 2 : Sequence Filtering

Given the set of candidate sequences $\mathcal{S}$ from the first step, we want to filter the sequences that fulfill our design criteria, i.e. form our given structure P as their unique global energy minimum (UGEM).

Since we are interested in energy optimal structure information, we apply the CPSP approach by *Backofen and Will* (2006) introduced in Sec. 3.2.1. Using this approach we can accomplish both: (a) calculating the optimal energy $E_{\min}(S)$ for a given sequence $S \in \mathcal{S}$ to check if the structure P is an optimal one (i.e. $E(S,P) = E_{\min}(S)$), and (b) to see if the sequence's degeneracy $\deg(S)$ is within the targeted limits (i.e. for a unique optimal structure: $\deg(S) = 1$).

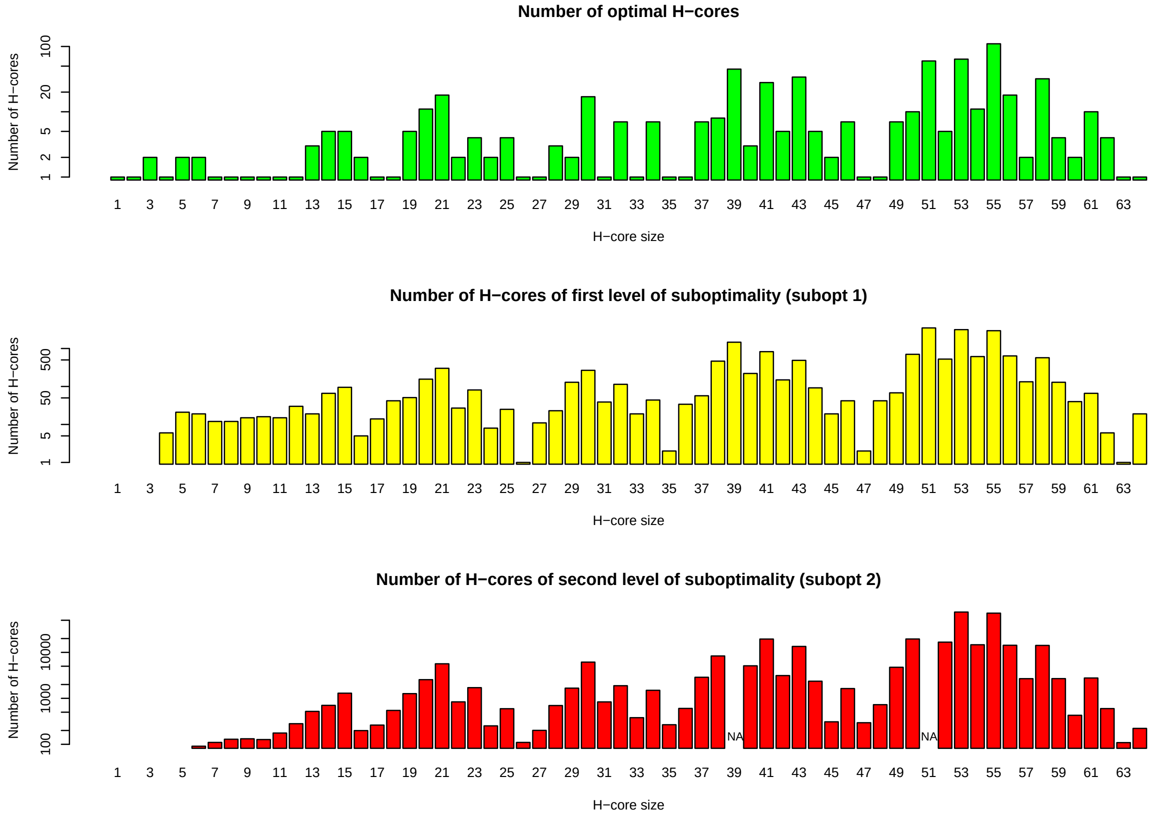


Figure 6.3: Numbers of optimal and suboptimal H-cores for various H-core sizes k in the 3D-cubic lattice.

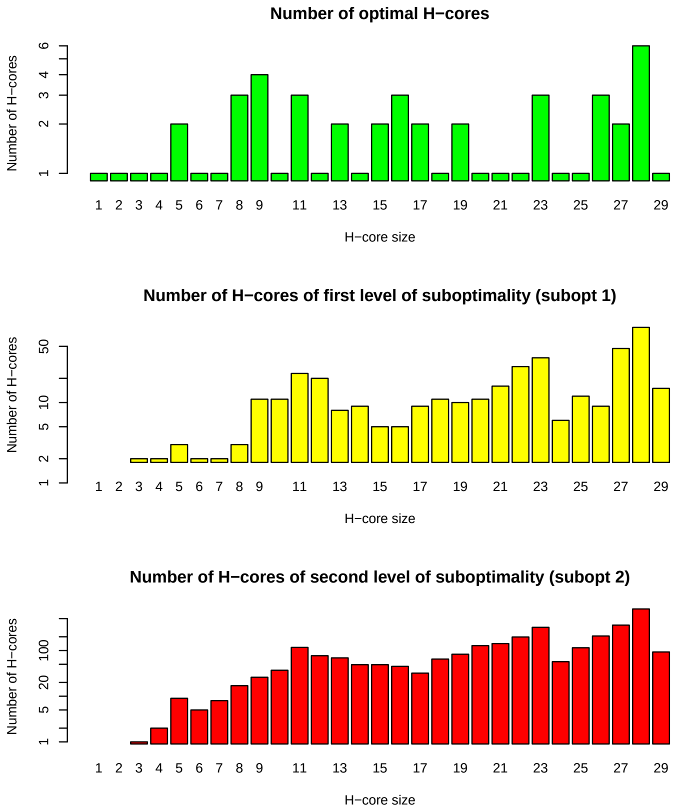


Figure 6.4: Numbers of optimal and suboptimal H-cores for various H-core sizes k in the 3D-face-centered-cubic (FCC) lattice.

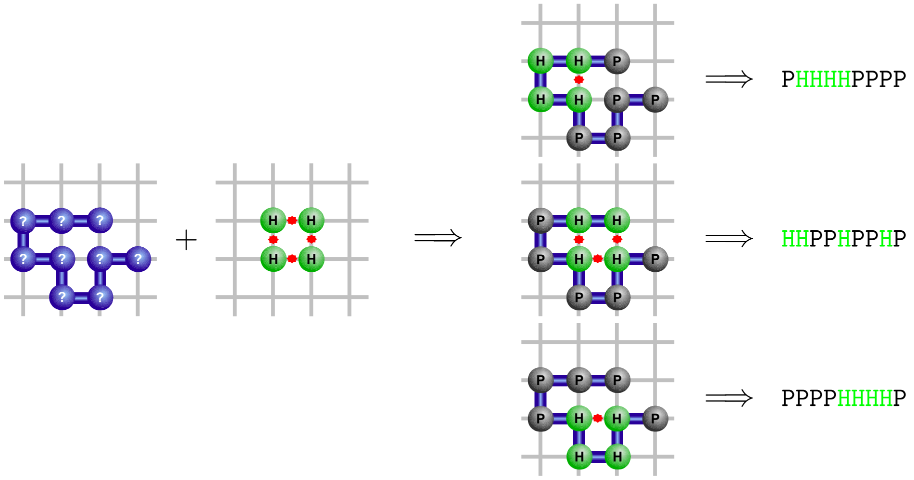


Figure 6.5: Sequence generation: For a given structure P (blue) in the 2D-square lattice and an optimal H-core $h \in \mathcal{H}_4$ of size 4 (green) all possible matches (middle) and derived sequences (right) are presented. These sequences are filtered in the next step (see Fig. 6.6).

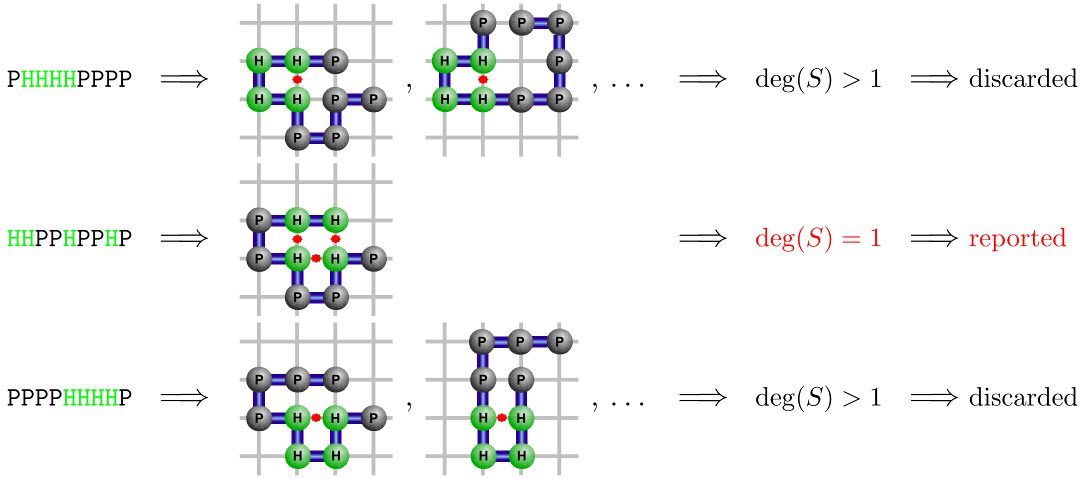

PHHHHPPPP ⟹
, , . . . ⟹ deg(S) > 1 ⟹ discarded
HHPPHPPHP ⟹
⟹ deg(S) = 1 ⟹ reported
PPPPHHHHP ⟹
, , . . . ⟹ deg(S) > 1 ⟹ discarded

Figure 6.6: Sequence filtering: The three candidate sequences from Fig. 6.5 are tested for their number of optimal structures in the 2D-square lattice and if the given structure P is among them. For each sequence up to two optimal structures are listed. Each sequence S can adopt P as an optimal structure (first structure listed), but only the second sequence forms it exclusively as its UGEM structure and thus has the targeted degeneracy $\deg(S)$ of 1.

For our study here, we are only interested in sequences forming P as their UGEM structure. Due to that, we can combine both tasks in the following way: For each sequence $S \in \mathcal{S}$, we enumerate up to 2 optimal (i.e. energy minimal) structures using the CPSP approach. If two structures are calculated, the sequence has a degeneracy of at least 2 and is therefore discarded. If only one optimal structure P' can be formed by S, we check if P' equals our target structure P. This check is done using a string comparison of the normalized absolute move string representations for both structures (see Sec. 1.2.2.3). In case of identity, the current sequence S under consideration was shown to form the target structure P as its UGEM structure and is reported.

As already mentioned above, one might want to relax the constraints on the sequence: For instance, one would like to ensure that the given structure P is formed as an optimal one but that the sequence's degeneracy is below a given threshold $\deg_{\max} \geq 1$. To this end, the following strategy can be applied. For each sequence $S \in \mathcal{S}$, we enumerate exactly one optimal structure P' using the CPSP approach. Via a comparison of the energies of both structures when formed by S, we can verify if P is an optimal structure. This holds if $E(S,P) = E(S,P')$. If we could show that P is an optimal structure, we can use the CPSP approach again to check the sequences degeneracy. Since we are only interested if the given degeneracy threshold $\deg_{\max}$ is exceeded, we restrict the degeneracy calculation via HPDEG (introduced in Sec. 3.3) to an upper limit of $\deg_{\max} + 1$, i.e. abort the calculation as soon as the limit is exceeded. In case HPDEG reports a degeneracy within the threshold ($\leq \deg_{\max}$), both criteria are fulfilled and the sequence is reported.

For the example illustrated in Fig. 6.5, only the second of the three derived candidate sequences shows a degeneracy of 1. The two other sequences (top and bottom) can form many more optimal structures due to the flexible P-monomer tails as depicted in Fig. 6.6.

6.2.2 Discussion

HPdesign enables the identification of sequences that form a given structure as their unique global energy minimum (UGEM) and thus solves the inverse folding problem. The method is based on the observation that optimal structures show an optimal or slightly suboptimal H-core and thus uses (sub-) optimal H-cores to derive a set of candidate sequences that have a high chance to be a solution for the problem. The fast and efficient CPSP approach is used in a second step to filter the candidate sequences for the final solutions.

Such a sequence space independent sequence design has a big advantage: the sequences predicted by HPdesign can be wide spread in sequence space. This is of importance when we want to study neutral networks since it was shown for 2D-models that such networks are often decomposed into several connected components (*Reidys et al.*, 2006; *Blackburne and Hirst*, 2003; *Noirel and Simonson*, 2008). Using HPdesign, we predict sequences from the neutral network for a given structure. Due to the structure focusing method, the computed sequences are quite likely from different connected components. This enables the study of neutral networks and their decomposition for complex 3D lattices without an exhaustive sequence space enumeration. In the next section such a study is given for the 3D-cubic lattice.

6.3 Neutral Network Exploration

As already discussed in the introduction, neutral networks are a common tool to visualize and study the parts of sequence space accessible to neutral evolution according to the neutral theory (*Schuster et al.*, 1994; *Bornberg-Bauer*, 1997b; *Reidys et al.*, 2006). Thus, they represent the sequences that encode a given function, usually defined via sharing the same unique global energy minimum (UGEM) structure. The edges of the network connect nodes of sequences that can be transformed into each other via mutations while usually only point-mutations are considered (*Xia and Levitt*, 2004a; *Wroe et al.*, 2005; *Noirel and Simonson*, 2008). Other mutation models like recombinations have been studied as well (*Cui et al.*, 2002).

So far, most neutral network studies are confined to the simple 2D-square lattice (*Bornberg-Bauer*, 1997b; *Wang et al.*, 2000; *Chan and Bornberg-Bauer*, 2002; *Wroe et al.*, 2007) or even a restricted version, where only structures with quadratic shape are considered (*Xia and Levitt*, 2004a; *Noirel and Simonson*, 2007, 2008). For an investigation of the qualitative differences between unrestricted and restricted 2D-square lattice models refer to (*Wroe et al.*, 2005). Studies in 3D-lattice models are less prominent in literature (*Bastolla et al.*, 2000; *Blackburne and Hirst*, 2003).

In the following, we will study neutral networks in unrestricted 3D-lattice HP-models. Here, we define the *neutral network for a structure* P as an undirected graph $\mathfrak{N}(P)$. Each node represents a sequence that forms P as its UGEM structure. The set of the represented sequences defines the *neutral set* $\mathfrak{S}(P)$. Two nodes are connected if the corresponding sequences differ only in one sequence position, i.e. can be transformed into another via a single point-mutation. A *connected component* is a maximal subgraph of $\mathfrak{N}(P)$ that contains a path between any two nodes within the component. Thus, a neutral network can be made of several independent connected components.

The central requirement for all sequences within the neutral set and network is to form a given structure P as their UGEM structure. As we have discussed in Chapter III, the prediction of optimal structures is a hard NP-complete computational problem even for simple 2D-lattices (*Berger and*

Leighton, 1998). Therefore, almost all studies so far are confined to short sequence lengths in order to solve the problem with an exhaustive structure space enumeration and evaluation, e.g. using length 18 in the 2D-square lattice (*Chan and Dill*, 1996; *Bornberg-Bauer*, 2002) which enables about 15 million possible structures (see Tab. 1.2 on page 11). In 3D-lattices, full structure space enumeration is not applicable anymore, at least for reasonable sequence lengths. Within 3D HP-models, fortunately we can apply the CPSP approach to calculate the optimal structures of a given sequence and thus solve the problem.

Within Section 6.3.1, we introduce the tool HPnnet that enables the neutral network expansion for HP-models in the 3D-cubic and 3D-FCC lattice. For a given structure P and a subset $\overline{\mathfrak{S}}$ of its neutral set $\mathfrak{S}(P)$ (where $\overline{\mathfrak{S}} \subseteq \mathfrak{S}(P)$ and $|\overline{\mathfrak{S}}| \geq 1$), the method recursively explores the neighbored sequence space of the given sequences $S \in \overline{\mathfrak{S}}$ in order to expand the connected components populated by $\overline{\mathfrak{S}}$. In Section 6.3.2, we combine the method with our sequence design tool HPdesign. The combination enables the expansion of the initial sequence set $\overline{\mathfrak{S}}$ using HPdesign such that sequences from different components of the neutral net $\mathfrak{N}(P)$ can be populated (see Sec. 6.2). This is necessary, since neutral networks usually decompose into several connected components (*Blackburne and Hirst*, 2003; *Wroe et al.*, 2005; *Noirel and Simonson*, 2008). Afterwards, all populated connected components are expanded using HPnnet. This protocol enables the study of neutral networks and their component decomposition without exhaustive sequence space explorations that are not applicable for longer sequence lengths. Thus, we present the first study of neutral networks in the unrestricted 3D-cubic HP-model for sequence lengths 27 and 36 in Sec. 6.3.2.

6.3.1 Approach

The neutral network approach HPnnet enables the expansion of neutral network components. To this end, a recursive breadth-first-search in sequence space is done following the point-mutation based edge-definition of the neutral networks studied. For each considered sequence, the CPSP approach is used to evaluate if the sequence is part of the neutral network or not.

The method requires as input a structure P and a subset $\overline{\mathfrak{S}} \subseteq \mathfrak{S}(P)$ of the structure's neutral set $\mathfrak{S}(P)$. As a result it produces a subgraph $\mathfrak{N}'$ of the neutral network $\mathfrak{N}(P)$ of P covering all connected components populated by $\overline{\mathfrak{S}}$. If the initial sequence subset $\overline{\mathfrak{S}}$ contains at least one sequence of each connected component, the produced graph will be the whole neutral network, i.e. $\mathfrak{N}' = \mathfrak{N}(P)$.

In detail, HPnnet maintains a set of sequences to process $\mathcal{S}$ and the set of already processed and tested sequences $\mathcal{T}$ to ensure that each sequence is considered only once. The set to process is initialized with the input sequences, i.e. $\mathcal{S} = \overline{\mathfrak{S}}$, the other set is empty, i.e. $\mathcal{T} = \emptyset$. Afterwards, the neutral network expansion follows a breadth-first search: We pick a sequence $S \in \mathcal{S}$ and shift it from the unprocessed to the processed set, i.e. $\mathcal{S} = \mathcal{S} \setminus \{S\}$ and $\mathcal{T} = \mathcal{T} \cup \{S\}$. The CPSP approach is used, to check if the sequence S has the given structure P as its UGEM structure. The procedure equals the filtering step of HPdesign described in Sec. 6.2.1.2. If the test holds, the sequence is part of the neutral set $\mathfrak{S}(P)$ of P and thus a node is added to the neutral network graph $\mathfrak{N}'$ and to the represented neutral set $\mathfrak{S}'$. Afterwards, the set of all neighbors $T(S)$ of S in sequence space according to point mutations is produced, i.e. n neighbors ($= |T(S)|$) for a sequence of length n. If a sequence $S' \in T(S)$ is already part of the neutral network graph $\mathfrak{N}'$, an edge (S, S') between the sequences is added to the graph. If no edge was added and S' was not already processed, i.e.

$S' \notin \mathcal{T}$, we add it for later processing to the queue, i.e. $\mathcal{S} = \mathcal{S} \cup \{S'\}$. When we have investigated all neighbored sequences $S' \in T(S)$, we iterate the whole procedure until no sequence is left in the candidate set $\mathcal{S}$. The resulting graph $\mathfrak{N}'$ will contain all connected components of the neutral network $\mathfrak{N}(P)$ that share a sequence with the initial sequence set $\overline{\mathfrak{S}}$. Thus, the set of all sequences $\mathfrak{S}'$ represented in $\mathfrak{N}'$ is at least as large as the initial neutral set $\overline{\mathfrak{S}}$, i.e. it holds $\overline{\mathfrak{S}} \subseteq \mathfrak{S}' \subseteq \mathfrak{S}(P)$.

6.3.2 Application and Results

In the following, we want to investigate the abundance and features of neutral networks in the 3D-cubic HP-model. We are interested in sequence lengths of at least 27. Here, a full enumeration of sequence space of more than $2^{27} \approx 134$ million sequences for an exhaustive study is not suitable. Therefore, we undertake a large randomized study to get sound statistics.

To generate the neutral networks for our study, we apply the following steps: (I) Identification of a large random set of non-degenerated sequences all over the sequence space and their UGEM structures, (II) calculation of parts of the neutral set for each UGEM structure from (I), and (III) expansion of the neutral networks starting from the sequence sets from (II). In the following, the steps are given in detail.

(I) Identification of uniquely folding sequences using HPoptdeg

Sebastian Will has introduced a Metropolis Monte-Carlo procedure to identify non- or low-degenerated sequences in 3D HP-models (*Will*, 2005). The method builds on the observation that there are degeneracy-gradients within sequence space that lead to low-degenerated sequences. Thus, the approach performs a local search in sequence space guided by a degeneracy-based Metropolis criterion. All degeneracy calculations are done using the CPSP approach (*Backofen and Will*, 2006). The starting point of the local search is a randomly picked sequence S_{start} with a relatively low degeneracy of $\deg(S_{\text{start}}) < 5,000$. For each following search step, we mutate the sequence S_{last} of the last step via one random point-mutation to derive the mutated neighboring sequence S_{mut} of S_{last}. Using the CPSP approach, the degeneracy $\deg(S_{\text{mut}})$ of the mutated sequence S_{mut} is determined. If the degeneracy of the mutated sequence is lower than the previous one's, i.e. $\deg(S_{\text{mut}}) < \deg(S_{\text{last}})$, the mutated sequence is accepted for the next iteration, i.e. $S_{\text{next}} = S_{\text{mut}}$. If the new degeneracy is higher, the mutated sequence is only accepted according to the Metropolis criterion using the probability

$$\Pr(S_{\text{last}} \to S_{\text{mut}}) = \exp\left(\frac{\ln(\deg(S_{\text{mut}})) - \ln(\deg(S_{\text{last}}))}{-T_s}\right)$$

where the simulation temperature T_s is set to 0.4. In case the mutated sequence is not accepted, the previous sequence is kept, i.e. $S_{\text{next}} = S_{\text{last}}$. The procedure iterates until a non-degenerated sequence S_{next} with $\deg(S_{\text{next}}) = 1$ is found or a maximal number of steps is exceeded. Here, we set search up to 10,000 steps. For further details on the method please refer to *Will* (2005).

We used the method's implementation HPoptdeg to generate a non-redundant set of non-degenerated sequences. For length 27 in the 3D-cubic lattice, we performed in total 19,724 HPoptdeg runs where 14,291 (73%) ended in a non-degenerated sequence. Removing duplicates, we derived 7,637 unique sequences that fall into 6,214 neutral sets, i.e. show 6,214 different UGEM structures. For length 36, 16,794 out of 78,555 runs (21%) yielded a non-degenerated sequence, but almost no duplicates were found. Overall, 16,684 unique non-degenerated sequences of length 36 out of 16,652 neutral sets we identified using HPOPTDEG.

In most cases only a single sequence from a neutral set was identified using HPOPTDEG. We found on average 1.23 ($\frac{7{,}637}{6{,}214}$) and 1.0 ($\frac{16{,}684}{16{,}652}$) sequences per neutral set for length 27 and 36, respectively. Assuming that a neutral network is spanned only in a small compact part in sequence space, this result shows the sampling via HPOPTDEG to be wide spread and unbiased towards single neutral networks.

(II) Sequence design for UGEM structures using HPdesign

The 6,214 and 16,652 UGEM structures of length 27 and 36, resp., are now used to design sequences from the respective neutral sets. To this end, we run HPDESIGN, described in Sec. 6.2, for each structure. This results in a subset of the neutral set for each structure. Since the sequences are designed without explicit sequence space search, they are likely to be part of different connected components of the according neutral network. This advantage is used in step (III) when expanding the networks.

We restrict HPDESIGN to generate and test on sequences with at least 40% H-monomers, i.e. 10 for length 27 and 14 for length 36. This is due to the observation that non-degenerated sequences tend to show about 50-60% H-monomers (data not shown). Furthermore, optimal and suboptimal H-cores up to the second level of suboptimality (see Sec. 6.2.1) are used to enable a wide screen in sequence space at length 27. For length 36, the search was restricted up to the first level of suboptimality.

Using HPDESIGN we were able to increase our initial sequence sets from step (I). For length 27, 1,798 new sequences were computed resulting in 9,435 sequences. Thus, we got on average 1.52 ($\frac{9{,}435}{6{,}214}$) sequences for each of the 6,214 neutral sets. At length 36, 5,336 additional sequences for the 16,652 UGEM structures were found using HPDESIGN resulting in 22,020 sequences, i.e. on average 1.32 sequences per neutral set ($\frac{22{,}020}{16{,}652}$).

(III) Neutral network expansion using HPnnet

Now we want to expand and study the neutral networks of our UGEM structures starting from the sequences found in step (I) and (II). To this end, we apply our HPNNET approach described in Sec. 6.3.1. As described there, HPNNET takes as input a structure P and a subset $\overline{\mathfrak{S}} \subseteq \mathfrak{S}(P)$ of the neutral set $\mathfrak{S}(P)$ of P, i.e. sequences that form P as their UGEM structure. Starting from the sequences within $\overline{\mathfrak{S}}$, HPNNET exhaustively enumerates the connected components of the neutral network $\mathfrak{N}(P)$ of P that are populated by sequences from $\overline{\mathfrak{S}}$. Thus, the resulting subgraph $\mathfrak{N}'$ of the neutral network represents an enlarged subset $\overline{\mathfrak{S}}'$ of the neutral set $\mathfrak{S}(P)$ of P.

For the 6,214 structures of length 27 we identified 9,435 sequences of their neutral sets using HPOPTDEG and HPDESIGN in step (I) and (II), respectively. The neutral network expansion via HPNNET revealed further 414 sequences part of the neutral sets of the known structures. In total, 9,849 sequences covering the neutral networks of 6,214 structures were identified. Given the expected fraction of about 0.015% for non-degenerated sequences, see Sec. 3.3.1, we would only expect $0.015\% \cdot 2^{27} = 20,133$ uniquely folding sequences of length 27. Therefore, our study covers already about 47% of all expected non-degenerated sequences of that length without a full enumeration of sequence

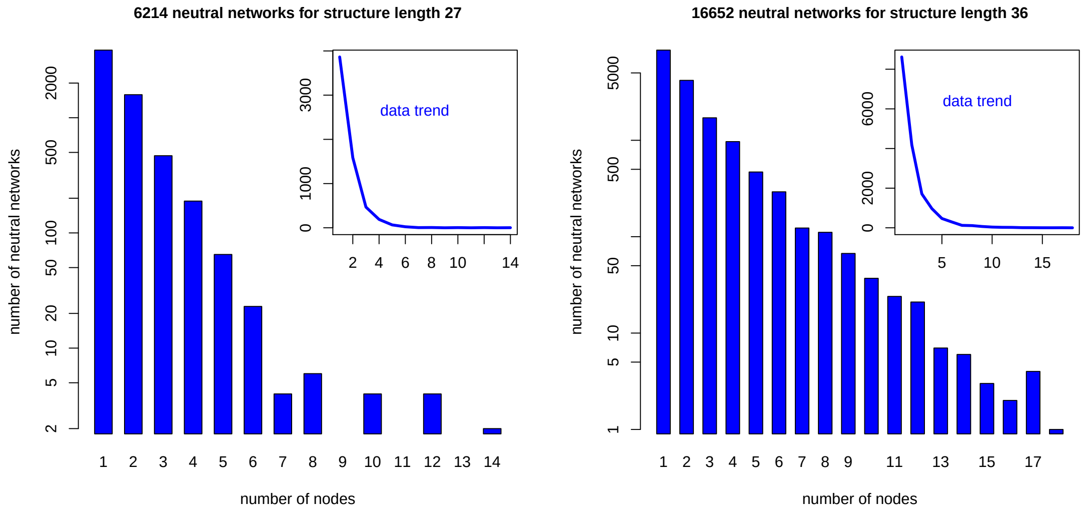


Figure 6.7: Node number distributions for neutral networks of structures with 27 and 36 monomers. The main plots show the logarithmic distributions while the insets depict the non-logarithmic data trends.

space. Furthermore, we discovered new neutral networks neighbored to the known ones during the network expansion. Using HPnnet, we found 420 sequences with so far unknown UGEM structure. By that, we identified additional 399 structures that sport a neutral network, but these networks were not considered further within this study to keep the presentation straight and clear.

At sequence length 36, 16,652 neutral networks were expanded based on 22,020 sequences. HPnnet identified 11,653 additional sequences part of these neutral networks resulting in a total number of 33,673 sequences. This large increase, compared to structure length 27, hints already that we do not have such a high coverage of non-degenerated sequences as for length 27. Only 0.33% of the expected 10.3 million non-generated sequences of length 36 are covered by the study. Nevertheless, the absolute numbers are large enough to enable sound statistics to check and support our findings from length 27 as we will show in the following section. As expected from the lower coverage of sequence space, HPnnet identified a large number of 6,468 unknown sequences sporting 5,971 so far unknown UGEM structures. Their neutral networks too have been excluded from the remaining study.

Properties of neutral networks in the 3D-cubic HP-model

In the following we will investigate the identified neutral networks from above. Thus, at sequence length 27 we consider 6,214 neutral networks covering 9,849 sequences and at length 36 we study 16,652 networks representing 33,673 sequences.

In Figure 6.7, we show the neutral network size distributions. The network sizes follow an exponentially decreasing trend, best visible for length 36 due to the larger data set. The maximal number of nodes for the networks studied is quite small, 14 for length 27 and 18 at length 36. When comparing the achieved neutral network sizes we see a length dependent increase. This is analogous to the results of *Blackburne and Hirst* (2003) for the simpler 3D-diamond lattice. But in contrast

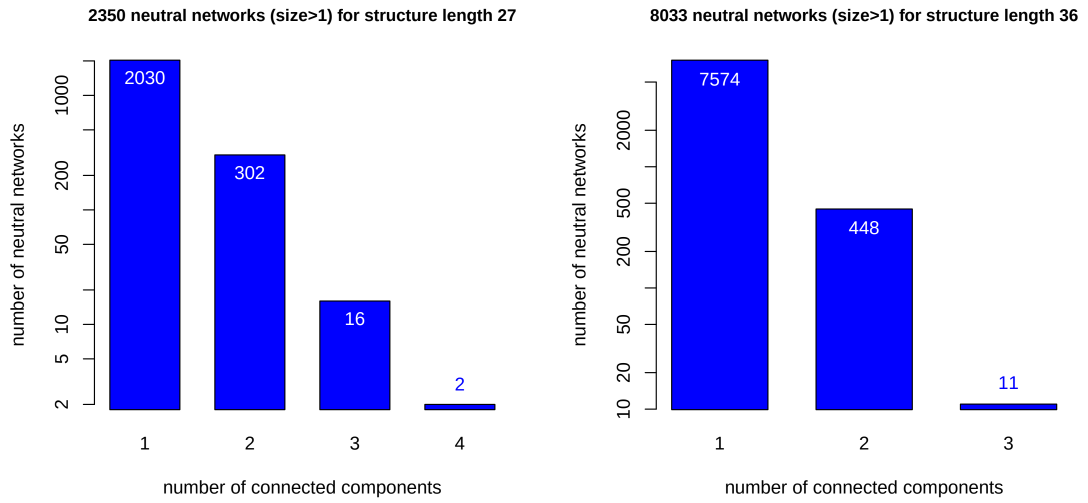


Figure 6.8: Connected component number distributions for neutral networks with at least two nodes for structures with 27 and 36 monomers in the 3D-cubic HP model.

to their results and to studies in the 2D-square lattice, we observe only very small network sizes on average. For instance, *Bornberg-Bauer* (1997b) found the largest neutral networks for much shorter structures of length 18 in the 2D square lattice to sport already 48 sequences. The reduced size of neutral networks in 3D-models results from the increased structural flexibility compared to 2D-models and the similarly simple 3D-diamond lattice (see Sec. 1.2.1). Already a few sequence mutations will result in a different optimal structure which is most probably not unique anymore. Still, the distribution follows a Zipf's law as reported for 2D models by *Bornberg-Bauer* (1997b) and which is known for RNA models too (*Grüner et al.*, 1996a,b). Accordingly, there are only a few structures formed by many sequences within a (relatively) large neutral network and many rare structures that are formed only by one or a few sequences (see insets in Fig. 6.7). Within the data, neutral networks show on average 1.6 and 2 sequences for length 27 and 36, respectively, while the median is in both cases 1 sequence.

Our finding for the 3D-cubic lattice are in contrast to the 2D-square lattice results by *Noirel and Simonson* (2008) who found that neutral networks usually contain one giant connected component and a number of very small ones. In our data we see on average only very small neutral networks or no large difference between component sizes. This means, that large neutral networks and thus evolutionary robust functions, i.e. structures, are rare in the 3D-cubic HP-model. On the other hand, this is not in contradiction to observations from nature where we find large sequence families for single folds or motifs. The model's abstraction of sequence space represents only the hydrophobic/polar characteristics of sequences. Thus, a single sequence in the HP model represents a set of sequences in the full 20-amino-acid alphabet. Therefore we conclude that small neutral networks in the HP model are models for larger networks in real sequence space. Note further that neutral network studies are restricted to a single sequence length while sequences for a single real fold show some variation in lengths. This problem is further discussed and investigated in Sec. 6.4.

As already mentioned above, larger neutral networks are likely to show several independent connected components. In Figure 6.8 the distribution of neutral networks with different numbers of connected components is presented. The plot is restricted to neutral networks that cover at least two nodes, i.e. 2,350 (of 6,214) for length 27 and 8,033 (of 16,652) for length 36. The majority of neutral networks shows a single component only. Nevertheless, a significant fraction is decomposed into several connected components.

In Figure 6.9, we give three examples of decomposed neutral networks for structure length 27. The given graphs are already large networks as we know from the node distribution given in Fig. 6.7. What is surprising, all three decomposed neutral networks can be joined into a single component if we allow for two point mutations (i.e. a Hamming distance of 2). In the figures such mutations are represented as dotted lines while single point mutations are given as solid lines within the neutral network graph.

Following this observation, we investigated all neutral networks with at least two independent components, i.e. 320 neutral networks for length 27 and 459 networks for length 36 (see Fig. 6.8). When connecting the graph with edges representing a maximal Hamming distance of 2, only about 20% of these networks are still decomposed (70 for length 27 and 89 for length 36). All these networks show only two connected components. When increasing to a maximal Hamming distance of 3, the fraction once more halves and drops to 46 and 28 neutral networks for length 27 and 36, respectively. Thus, only a tiny fraction of less than one percent of all neutral networks ($\frac{46}{6{,}214}/\frac{28}{16{,}652}$) shows a Hamming distance larger than three between its independent components, while their median minimal component distance is 4 (mean $\sim$ 4.5) in both data sets. From this it follows that the majority of neutral networks is quite compact in sequence space. Thus we presume that most functions, i.e. stable structures, are connected to specific sequence features.

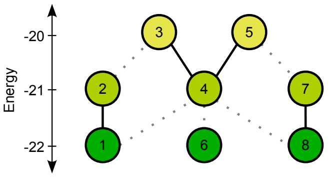
Distribution of neutral networks in sequence space

Within the last section, we have investigated properties of individual neutral networks and the distribution of their independent components in sequence space. Now, we look into the distribution of all these neutral networks in sequence space and their relations to each other.

To this end, we take all sequences from all neutral sets and generate a single pseudo neutral network covering all individual neutral networks. Thus for structure length 27, the graph contains 9,849 sequence nodes. As for standard neutral networks, we connect all nodes with according sequences' Hamming distance of 1 and calculate the number of independent connected components within the graph. The graph is decomposed in 6,017 components and thus resembling the inherent number of individual neutral networks of 6,214. As we have shown for individual neutral networks, the connectivity might be much higher when considering more point mutations, i.e. a Hamming distance of 2 or higher. When doing so, the number of connected components for structure length 27 drops from 6,017 to 1,398 to 343 to 41 to 1 for increasing Hamming distance from 1 to 5. That is at Hamming distance 5, the graph contains only a single component containing all sequences. When investigating the size of the largest components we discover the growth of a single giant component within the graph. For Hamming distance 2 the largest component covers already 75% of all 9,849 sequences which is at Hamming distance 3 already increased to 95%. Note, the connectivity of neutral networks might be even higher since our study does not cover all neutral networks but only a large fraction (see step III of the neutral network expansion above).

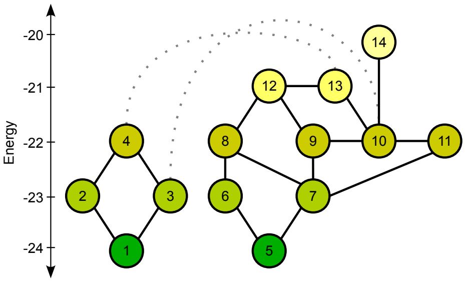
Note, the high connectivity at Hamming distance larger than 4 is expected when considering

a) $P =$ FRRBBUFFLUBDDDBULFLFRUBBRRF



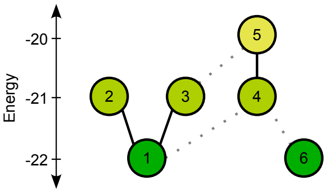
node	S	$E(S,P)$
1	HHHHHPHHHHPHHHPHPHPPHPHPHHH	-22
2	HPHHHPHHHHPHHHPHPHPPHPHPHHH	-21
3	HPHPHPHHPHPHHHPHPHPPHPHPHHH	-20
4	HHHPHPHHPHPHHHPHPHPPHPHPHHH	-21
5	HHHPHPHHPHPHHHPHPHPPHPHPHPH	-20
6	HHHPHPHHPHHHHHPHPHPPHHHPHHH	-22
7	HHHPHPHHPHPHHHPHHHPPHPHHHPH	-21
8	HHHPHPHHPHPHHHPHHHPPHPHHHHH	-22

b) $P =$ FRUULDBURRFFLLDDRRBBLURFFL



node	S	$E(S,P)$
1	HHHHHPHHPHHHPHPHHHPHHHHHHPH	-24
2	HHHHHPHHPHHHPHPHPHPHHHHHHPH	-23
3	HHHHHPHHPHPHPHPHHHPHHHHHHPH	-23
4	HHHHHPHHPHPHPHPHPHPHHHHHHPH	-22
5	HHHHHPHHPHPHHHPHHHHHPHHPHHH	-24
6	HHHHHPHHPHPHHHPHPHHHPHHPHHH	-23
7	HHHHHPHHPHPHHHPHHHPHPHHPHHH	-23
8	HHHHHPHHPHPHHHPHPHPHPHHPHHH	-22
9	HHHHHPHHPHPHPHPHHHPHPHHPHHH	-22
10	HHHHHPHHPHPHPHPHHHPHPHHPHPH	-22
11	HHHHHPHHPHPHHHPHHHPHPHHPHPH	-22
12	HHHHHPHHPHPHPHPHPHPHPHHPHHH	-21
13	HHHHHPHHPHPHPHPHPHPHPHHPHPH	-21
14	HHHHHPHHPPPHPHPHHHPHPHHPHPH	-20

c) $P =$ FRFUBUBRDFDBBUULLFFDBBRDFU



node	S	$E(S,P)$
1	HHHPPHPHHHHHHPHHHPHPHHPHPHH	-22
2	HHHPPHPHHHHPHPHHHPHPHHPHPHH	-21
3	HPHPPHPHHHHHHPHHHPHPHHPHPHH	-21
4	HHHPPHPHPHHHHPHPHPHPHHPHPHH	-21
5	HPHPPHPHPHHHHPHPHPHPHHPHPHH	-20
6	HHHPPHHHPHHHHPHPHPHHHHPHPHH	-22

Figure 6.9: Examples of large neutral networks in the 3D-cubic HP-model of length 27. Each graph (a-c) represents the identified part $\mathfrak{N}'$ of the neutral network $\mathfrak{N}(P)$ of the given structure P (see text for details). The nodes are colored according to the energy $E(S,P)$ when the node's corresponding sequence S (see list on the right) forms P. Nodes connected by solid lines can be transformed into each other by point-mutations. Dotted gray lines show transformations between connected components that require only two point-mutations.

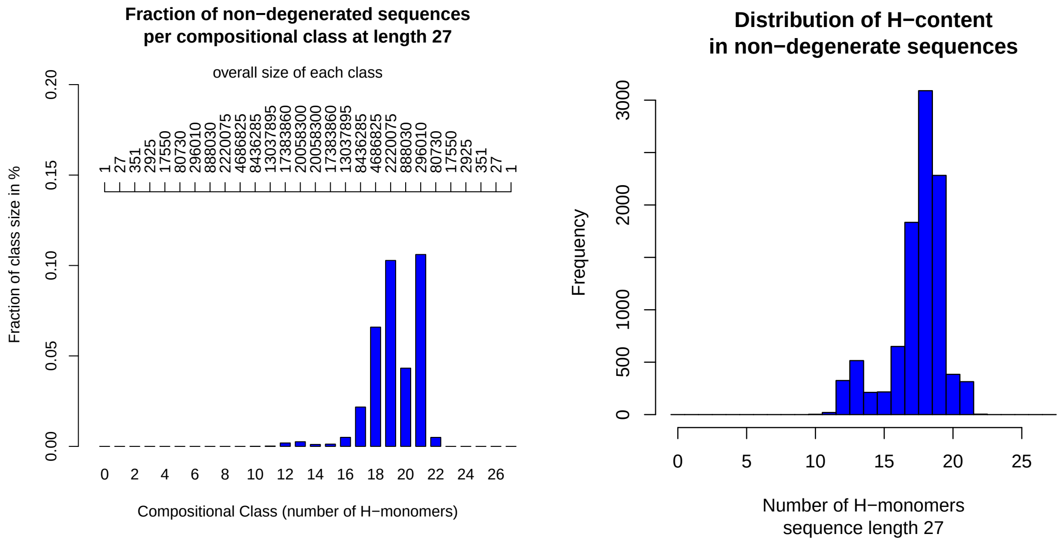


Figure 6.10: (left) Decomposition of the sequence space for length 27 into compositional classes concerning the number of H-monomers within a sequence and the fraction of the tested 9,849 non-degenerated sequences per class in the 3D cubic lattice. (right) Absolute histogram of the compositional class distribution for the same data.

the following prospect. Assume we sample x sequences of length 27 uniformly distributed over the whole sequence space of size $2^{27} \sim 130 \cdot 10^6$. For each sequence, we collect the surrounding sphere of neighbored sequence at Hamming distance k each comprising $\sum_{i \leq k} \binom{27}{i}$ sequences. Assuming our $x = 10,000$ sequences, we get already about $33 \cdot 10^6$ sequences for $k = 3$ and about $210 \cdot 10^6$ using $k = 4$. Thus we see that for $k \geq 4$ we start to get a multiple coverage of the whole sequence space, which comprises only $130 \cdot 10^6$ sequences. This enables the connectivity of all sampled sequences. For length 36 the coverage for Hamming distance 4 requires $x > 10^6$ sequences which is in according with the expected number of $10.3 \cdot 10^6$ non-degenerated sequences (0.015% of 2^{36}). For increasing sequence lengths n, we expect a decreasing minimal Hamming distances $k_{\min}$ for the coverage when focusing on the fraction of non-degenerated sequences only. In this case, we have to find a $k_{\min}$ such that

$$\frac{2^n}{1.5 \cdot 10^4} \sum_{i \leq k_{\min}} \binom{n}{i} \quad > \quad 2^n \tag{6.1}$$

$$\sum_{i \leq k_{\min}} \binom{n}{i} \quad > \quad 1.5 \cdot 10^4 \; . \tag{6.2}$$

For sequence length $n = 45$, the minimal Hamming distance k that solves Eq. 6.2 is already $k = 3$ but relies on our estimate that only 0.015% of the sequence space is non-degenerated. Since the estimate is based on observations up to sequence length 36, the ratio might be slightly higher resulting again in a minimal Hamming distance of $k = 4$.

Given the unexpected high connectivity of the neutral networks at a low Hamming distance of 2 or 3 for sequence length 27, we have to check if this observation is a significant feature of our data or an artifact of the populated sequence subspace. To this end, we assume as the null-hypothesis that our sequences are distributed randomly and independently within the sequence subspace. Since the latter does not hold for sequences from the same neutral set, we consider only a single randomly selected sequence per neutral set. For structure length 27, we end up with a set of 6,214 sequences. For these sequences, we generate the pseudo neutral network as done above for different Hamming distances ranging from 1 to 5. For comparison, we need to generate an according random graph. Since we observe that non-degenerated sequences show characteristic H/P ratios (see Fig. 6.10), we want to ensure that the random model resembles the same H/P ratios as present in the neutral set derived graph. Therefore, we generate a set of 6,214 random sequences by an independent shuffling of each sequence from the neutral-set-based graph. The resulting randomized graph is connected accordingly for increasing Hamming distances. The number of independent connected components as well as the size of the largest component is analysed for each maximal Hamming distance allowed for connections. The whole experiment is repeated 10 times to enable statistics.

Figure 6.11 presents the distributions of component numbers and largest component sizes for both the original (green) and the randomized graph (red). The boxes that represent 50% of each distribution are almost not visible revealing the very low variation in the data. Furthermore, the original data taken from the neutral sets (in green) shows significantly less connected components for Hamming distance 2 to 4 compared to the randomized data (in red). Furthermore, we see the growth of a single giant component within the original data for low Hamming distances that is only observed for larger distances of at least 4 within the shuffled data. The latter distribution is observed for any randomly selected sequence sets, i.e. at Hamming distance 4 to 5 the majority of sequences is connected in a large cluster, and is the expected threshold as discussed above. Consistently we see no giant component growth for a lower Hamming distance of 2 or 3 in randomized data. Only our neutral set derived data shows the giant component observed for our complete neutral set sequences described above. Therefore, we consider the high low-distance connectivity of neutral sets and the formation of a single giant component in sequence space a non-expected observation specific to our non-degenerated sequences. We observe a similar behaviour in the 2D-square lattice taking the according “designing sequences” and structures of length 18 from (*Irbäck and Sandelin*, 2000).

6.3.3 Discussion

Neutral networks are a central concept to study the extent and impact of neutral evolution in sequence space. They represent the accessible sequence space that can compensate mutations while preserving a specific structure and function. Many studies have investigated simple lattice models mainly in the 2D-square lattice (*Bornberg-Bauer*, 2002; *Wroe et al.*, 2005; *Bloom et al.*, 2007; *Noirel and Simonson*, 2008) but no exhaustive investigations have been done for the unrestricted 3D models. Beside the work of *Will* (2005), to our knowledge only *Bastolla et al.* (2000) studied the unrestricted 3D-cubic lattice. Other studies confine the structure space to maximally compact structures that form e.g. a 3×3×3 cube (*Hao and Scheraga*, 1996; *England and Shakhnovich*, 2003).

Within this chapter we have identified, expanded, and studied a large number of neutral networks in the unrestricted 3D-cubic lattice for different sequence lengths. Our data shows significant differences compared to 2D-square lattice results. Neutral networks in the 3D-cubic lattice comprise only about 0.02% of the sequence space compared to about 2.4% for 2D models (*Chan and Dill*, 1996). Neutral sets are much smaller and we do not see the existence of a single giant component for

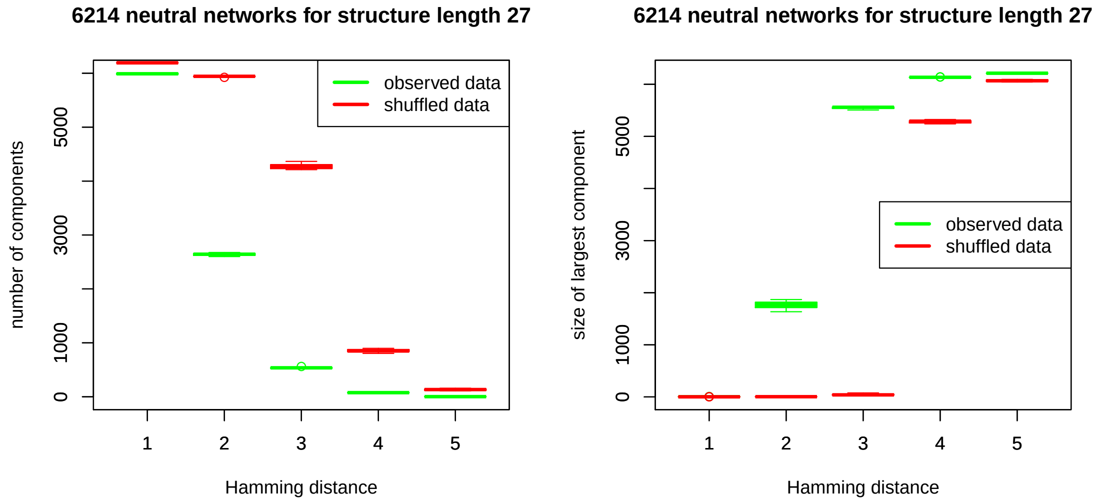


Figure 6.11: Distributions of the number of connected components (left) and the size of the largest component (right) for different Hamming distances (1-5). The analysis is based on graphs where nodes represent a random sequence from each neutral set for structure length 27 (green data) in the 3D-cubic lattice. Two nodes are connected if their sequences have a Hamming distance below the according threshold. The data is compared to a randomized version of the graph (red data) where each sequence is shuffled but its H/P ratio is kept the same. The procedure is iterated 10 times.

multi-component neutral networks, both known for 2D. Nevertheless, we see a similar distribution for the sizes of neutral sets following a Zipf's law.

Beyond that, we observe on average a very low Hamming distance of 2 between independent connected components of multi-component networks. Thus, neutral sets are located relatively compact within sequence space and are not randomly distributed. Furthermore, we studied the distribution of all our neutral networks within sequence space. Here, we found as well that we can connect the majority of neutral networks into one large connected component already at Hamming distance 2. For increasing Hamming distances, the component grows till it covers all neutral sets at distance 5. We could show that the latter is expected for Hamming distance larger than 4. But the growth of the giant component already at distance 2 was shown to be a surprising property of our neutral sets.

These results disprove the speculations drawn by *Bastolla et al.* (2000). They assumed neutral networks to be well separated in sequences space but used only stochastic simulations for their investigations. On the other hand, our results are extending the observation of *Will* (2005). He found that some neutral networks can be connected via low degenerated structures to larger clusters. Our investigations show, while ignoring the degeneracy of the linking sequences, that such interlinks are common and connect the majority of neutral networks in sequence space.

Based on our findings we see that evolution can access the majority of stable structures and functions without traversing (many) sequences with unfavorable, degenerated structures. When neutral evolution passes through a neutral network of a specific function, some mutations or mutation combinations will enable the transfer into another neighbored neutral network sporting a different function. Whether this new function will be maintained over time depends on the size of the neutral set of that structure, i.e. how many neutral mutations are possible, and the mutation rate. Furthermore, the entered neutral network might be the spring board to another function and so forth. These observations are in accordance with the evolutionary models postulated by *Wagner* (2008) and *Tokuriki et al.* (2008). Similar results have been reported for RNA as well (*Grüner et al.*, 1996a,b).

6.4 A Fold Definition for HP-models

Real protein structures often show structural patterns mandatory to perform a certain function (*Alberts et al.*, 1994). Due to that, they can be grouped into structural classes according to similar topological configurations of their backbones (*Pascarella and Argos*, 1992a; *Chothia*, 1992; *Holm et al.*, 1992). The common general patterns are called *folds*, or *structural motifs*, and can cover whole proteins or well-defined sub-domains of a structure. Since there is no straightforward relationship between a protein's sequence and structure, two proteins might share a fold but have very dissimilar sequences. Known examples of protein folds are the *helix-loop-helix*, *beta-alpha-beta*, or *beta-hairpin-beta* motifs shown in Fig. 6.12.

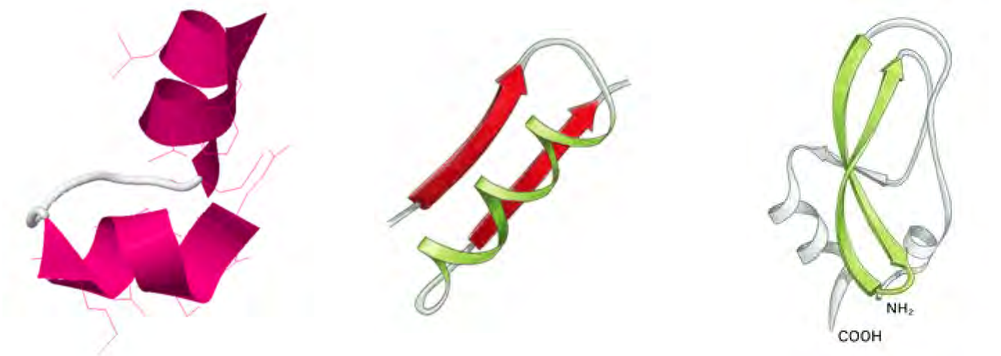


Figure 6.12: Examples of common structural motifs/folds in protein structures. (left) a *helix-loop-helix* motif, (middle) *beta-alpha-beta* motif, and (right) *beta-hairpin-beta* motif (Figures are taken from (*Alberts et al.*, 1994)).

The classification and grouping of proteins according to their folds is central to our understanding of protein function and evolution (*Ramnarayan et al.*, 2008). Proteins from the same class form similar folds and render an image of the evolutionary processes that shaped these proteins. Possible events are the mutation of single amino acids or their insertion and deletion, even of whole protein parts (*Pascarella and Argos*, 1992b).

In order to study the evolution of sequences for a given structural motif within HP-models, we first have to define such motifs or folds within our model. To our knowledge, no such definition exists in literature. Within this chapter, we are going to present such a definition and the algorithms to identify and predict the according motif for a given structure. To this end, we use the following assumptions that have to be met by the definition.

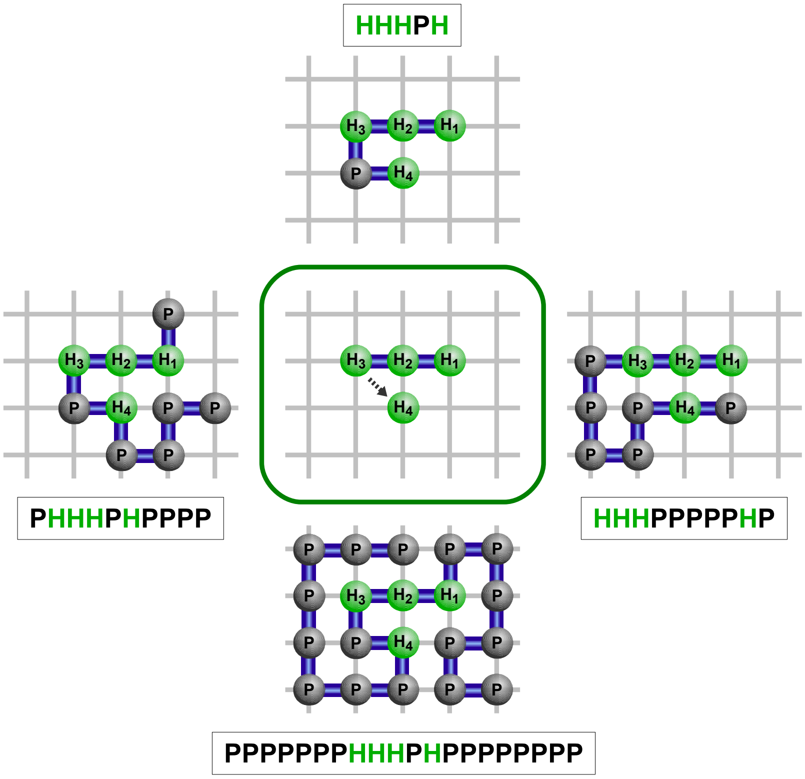


Figure 6.13: Different sequences and structures forming the same *H-fold* given in the center. Note, the H-fold is insensitive to the length of the connecting P-monomer stretches.

A structural motif . . . is a sub-domain of a protein structure.
. . . has to be thermodynamically stable to support a function.
. . . is independently described from the rest of the structure.

The structural motif description from above demands for a thermodynamically stable structure. Therefore, only energetically optimal structures are considered in the following. Since the HP-model energy calculation only accounts for hydrophobic interactions in the form of direct H-monomer contacts, its thermodynamic stability is defined by the protein's hydrophobic fragments only. The connecting P-monomer sub-structures, interlinking the fragments, are not constrained by the energy function. Thus, they form flexible parts in optimal lattice protein structures. Only the structure space discretization due to the underlying lattice might present some constraints on the P-monomer placement.

Based on these observations, we define the *H-fold* of a lattice protein structure in the HP-model. An H-fold encodes all hydrophobic fragments of the structure and their relative arrangements. Thus, the H-fold is independently defined from the flexible polar stretches of the structure as shown in Fig. 6.13. Since only the relative orientation of the H-monomer fragments is of importance, the length and placement of interconnecting P-monomer loops is not relevant. In the following, we will give a precise definition of an H-fold and how to obtain its unique string representation.

6.4.1 H-fold Definition

Given a backbone-only lattice protein (S, P) of length n with sequence $S \in \{\mathtt{H},\mathtt{P}\}^n$ and its structure $P \in L^n$. We describe its H-monomer fragments via the set $I_H(S)$ of hydrophobic interval boundaries within the sequence S given by

$$I_H(S) = \{ \ (i,j) \mid (\forall_{i \leq x \leq j} : S_x = \mathtt{H}) \ \wedge \ (i = 1 \vee S_{i-1} = \mathtt{P}) \ \wedge \ (j = n \vee S_{j+1} = \mathtt{P}) \ \}. \tag{6.3}$$

Now, we can extract the substructures of all hydrophobic fragments defined by $I_H(S)$. The sequence of these substructures ordered by their occurrence within the protein defines the H-fold $F(S, P)$ of a the protein structure (Eq. 6.4).

$$F(S, P) = (\ (P_i, \ldots, P_j) \mid \exists (i,j) \in I_H(S) \ ) \tag{6.4}$$

A graphical example of the definition is presented in Fig. 6.13. Here, all structures show a fold of two segments. Choosing the protein at the top of the figure, we have $S = \mathtt{HHHPH}$ and for instance a structure $P = \left(\ \binom{3}{2}, \binom{2}{2}, \binom{1}{2}, \binom{1}{1}, \binom{2}{1} \ \right)$. Therefore, the set of hydrophobic fragments is given by $I_H(S) = \{(1,3), (5,5)\}$ and the H-fold is $F(S, P) = (\ (\binom{3}{2}, \binom{2}{2}, \binom{1}{2}), \ (\binom{2}{1}) \ )$. The same H-fold would be obtained for the other sequences as well, given that the H-monomers are positioned exactly the same.

But what if we compare the H-fold of a structure to a rotated or translated version of itself? Since our H-fold definition is currently based on absolute coordinates, the H-folds would have to be superpositioned first or otherwise would be assumed to be different. That is not in accordance with the requirements for the description of a structural motif as given above. Thus, we need a unique descriptor of an H-fold based on a relative rather than an absolute positioning. Such a descriptor based on a string representation of an H-fold is introduced next.

6.4.2 Unique String Representations of H-folds

To obtain a unique string representation of an H-fold, we follow the idea of the move string normalization presented in Sec. 1.2.2.3. To this end, we encode the relative positioning within the H-fold as a move string and identify the symmetry that produces the lexicographically smallest move string. This string is unique and will serve as the string representation $\mathcal{F}$ of the H-fold.

In detail, given a protein (S, P) and its H-fold $F(S, P)$ showing $\eta = |I_H(S)|$ hydrophobic segments, where $(i_k, j_k) \in I_H(S)$ defines the kth segment in $F(S, P)$. The string is built in a sequential manner based on the segments. For each segment, a comma separated string representation is built and encapsulated in brackets "[...]". The resulting segment strings are connected by a string encodings of the distance vectors between the according segment ends. Using the string concatenation operator $\oplus$, we can construct a string representation of $F(S, P)$ with the following algorithm (Alg. 4) given the function VectorToString(...) (Alg. 1 on page 13) that converts a distance vector into a unique string based on 3D-cubic move encodings.

Since our string representation of an H-fold is basically an extended move string, we can apply the idea for move string normalization presented in Sec. 1.2.2.3 to generate a unique/normalized H-fold representation.

Algorithm 4 Algorithm to obtain a string encoding for a given H-fold.

```
function GetString($F(S,P)$)                    ▷ generates a string representation for an H-fold
    str ← SegmentToString($i_1, j_1$)           ▷ initialization with first segment
    for ($1 < k \leq \eta$) do                  ▷ add for each subsequent segment
        str ← str ⊕ VectorToString($P_{i_k} - P_{j_{(k-1)}}$)   ▷ segment connection
        str ← str ⊕ SegmentToString($i_k, j_k$)   ▷ current $k$th segment
    end for
    return str
end function

function SegmentToString($i, j$)      ▷ returns a string representation of an H-fold interval
    seg ← '['
    for ($i < x \leq j$) do                     ▷ comma separated list of relative positions
        seg ← seg ⊕ VectorToString($P_x - P_{x-1}$)   ▷ encoding of monomer connection
        if ($x \neq j$) then
            seg ← seg ⊕ ','                     ▷ add separator if needed
        end if
    end for
    seg ← seg ⊕ ']'
    return seg
end function
```

Given the string representation of the current H-fold ς = GetString($F(S,P)$) and the set of all possible symmetry functions $\mathcal{R}$ within the 3D-cubic lattice. Each symmetry function $r \in \mathcal{R}$ maps the move encoding of the neighboring vectors onto their corresponding encodings according reflection or rotation (see Sec. 1.2.2.3). Thus, we want to identify the symmetry $\check{r} \in \mathcal{R}$ that produces the lexicographically smallest H-fold string representation. This means in detail, given the current H-fold string representation $\varsigma = (\varsigma_1, \ldots, \varsigma_l)$ of length l, we enforce

$$\forall\, r \in \mathcal{R} \quad : \quad \mathrm{sym}(\varsigma_1|\check{r}) \oplus \ldots \oplus \mathrm{sym}(\varsigma_l|\check{r}) \;\; \leq_{\mathrm{lex}} \;\; \mathrm{sym}(\varsigma_1|r) \oplus \ldots \oplus \mathrm{sym}(\varsigma_l|r)) \tag{6.5}$$

$$\text{with} \tag{6.6}$$

$$\mathrm{sym}(\varsigma_i|r) \quad = \quad \begin{cases} \varsigma_i & \text{if } \varsigma_i \in \{\texttt{'['}, \texttt{','}, \texttt{']'}\} \text{ and} \\ r(\varsigma_i) & \text{otherwise.} \end{cases} \tag{6.7}$$

The lexicographically smallest H-fold representation based on symmetry $\check{r}$ gives us the *unique H-fold string encoding* $\mathcal{F}(F(S,P))$, i.e.

$$\mathcal{F}(F(S,P)) = \mathrm{sym}(\varsigma_1|\check{r}) \oplus \ldots \oplus \mathrm{sym}(\varsigma_l|\check{r}) \; . \tag{6.8}$$

We exemplify the procedure for the example from above. The H-fold from Fig. 6.13 is given by $F(S,P) = (\; (\binom{3}{2}, \binom{2}{2}, \binom{1}{2}),\; (\binom{2}{1}) \;)$. The corresponding H-fold string representation would be `[B,B]FL[]` since on the X-axis -1 moves are encoded by `B`, +1 moves by `F` and a Y-axis move of -1 by `L`. Since the second segment contains only one monomer, no relative positioning information is needed and it is represented by `[]`. A clockwise 90° rotation of the H-fold would result in the string `[R,R]BL[]`, given that `R` encodes +1 moves on the Y-axis. This exemplifies the ambiguity of the string representation without normalization. The normalized unique string representation of the presented H-fold is `[B,B]FD[]`. The method is implemented and available when calling HPrep with given sequence and structure, CPSP-tools v2.4.5 (*Mann et al.*, 2008b).

6.4.3 Analogy of H-folds and Equivalence Classes

In Sec. 3.4, we have introduced an equivalence relation $\overset{H}{\sim}$ on backbone-only lattice protein structures that ignores the P-monomer placements. For a given sequence S, two structures P, P' formed by this protein are considered equivalent iff they show the same H-monomer placement ignoring symmetries.

This definition exactly corresponds to the H-fold definition from above for a fixed sequence S. The same H-fold corresponds to the same arrangement of the `H`-monomer segments. Thus, the two structures of a sequence S are equivalent according to Eq. 3.4 if and only if they show the same H-fold string encoding.

$$P \overset{H}{\sim} P' \Longleftrightarrow \mathcal{F}(F(S,P)) = \mathcal{F}(F(S,P')) \tag{6.9}$$

By that, we directly get a machinery to enumerate the optimal H-folds for a given sequence by the enumeration of a representative for each equivalence class and the successively generation of the according H-fold for each representative. Note, an *optimal H-fold* is an H-fold shown by a structure with minimal energy for a given sequence. Therefore, the number of optimal H-folds a sequence can form corresponds to its core-degeneracy and each optimal H-fold describes an equivalence class of optimal structures. Therefore, we have already some statistics on the distribution of the number of optimal H-folds within sequence space via our equivalence class study from Sec. 3.4.2.

Revisiting the core-degeneracy study from Sec. 3.4.2 for sequence length 27 in the 3D-cubic lattice, we see a mean core-degeneracy of $4,856$, i.e. the average number of optimal H-folds for random sequences. From the one million random sequences tested, only 284 show a unique optimal H-fold, i.e. a core-degeneracy of 1. But when considering sequences with up to 10 optimal H-folds, their fraction comprises already about 4% of the tested sequences. Thus, a relatively large number of sequences can adopt a unique stable H-fold.

6.4.4 Discussion

Within real protein structures we can observe common structural patterns or motifs that are specific to a certian function. In agreement with the neutral theory of evolution, we often see a large variation within the protein sequences that adopt the same fold. But the differences within the sequences are not restricted to accumulated amino acid substitutions as studied via neutral networks in Sec. 6.3. Sequences that form the same structural motif often also differ in length which is not covered by the neutral network studies that are fixed to a single structure. The length variation is possible when the segments of the folds are interlinked or flanked with flexible loop/tail regions that often accumulate the majority of the neutral mutations (see Fig. 6.12). *Pascarella and Argos* (1992b) have studied insertions and deletions (indels) in real proteins. They found indels to be short and distributed all over the loop regions of protein structures while the interruption of α-helices and β-strands (forming the fold) are rare events.

These results show that the study of neutral evolution only based on amino acid substitutions within a fixed structure does not render a clear picture of the underlying processes. To enable reasonable indel studies, one first has to define a notion of folds within the used protein model to distinguish between mutation accessible loop regions and the fixed parts of the fold. Within this section, we have introduced such a definition for the HP model. The defined H-folds describe the fixed hydrophobic segments of optimal protein conformations and their relative arrangements to each other. Thus, an H-fold represents the parts of the protein constrained by the HP energy function that focuses on hydrophobic forces only. The interconnecting loop regions are not part of the H-fold description and can be of any size and shape.

We have introduced an efficient unique string encoding of H-folds to enable a fast extraction and comparison of H-folds for given structures. Furthermore, we presented the relationship to the hydrophobicity-focusing equivalence relation introduced in Sec. 3.4. Using the approach introduced for equivalence class representative enumeration, we can enumeration the optimal H-folds accessible for a given structure. Thus we see that a large fraction of already 4% of all HP sequences of length 27 show a low number of less than ten optimal H-folds.

Currently, we have started additional evolutionary studies based on our introduced H-folds that are similar to the neutral network explorations we have presented in Sec. 6.3. The usage of H-folds enables the investigation of loop evolution happening via insertion or deletion of amino acids within the flexible protein parts that connect the fixed hydrophobic fragments described by H-folds. Thus, instead of studying sequence evolution for a fixed structure, as done for neutral network analyses, we will study the sequence space accessible to neutral evolution for a fixed H-fold. But this is ongoing work and wont be presented within this thesis anymore.

CHAPTER VII

Conclusions

The study of proteins is central to broaden our understanding of the cellular processes that are omnipresent and essential within living creatures. Such investigations have deep impact and enable innovations in all fields of research ranging from medical science via chemistry to biotechnology.

Research in atomic detail is currently often only possible for a few specific questions. For studies requiring large scale analyses usually more abstract models are needed. These enable a good trade-off between computational complexity and the expressivity of the conclusions drawn. Within this thesis, we focused on lattice protein models that discretize the possible structure space of proteins. Due to that, fast folding simulation protocols as well as the investigation of features of the structure space are possible. The according thermodynamics are ruled by the different energy functions applied.

Within this manuscript, we have investigated backbone-only and side chain lattice protein models in 3D lattices. New methods and approaches have been introduced that extended the computational accessibility of the models and enabled new studies of their properties. All methods have been successfully implemented and are available in two major software packages as well as via a web frontend for future research based on this thesis. The theoretical basis as well as the introduced methods in concert with several studies and applications were presented within each chapter.

Chapter I gave a general introduction into protein research and the abstractions that define the studied lattice protein models. Here, we did first studies of the growth of their structure space in different lattices and gave length-dependent estimators. It became clear that the structure space of side chain models shows a vast exponential growth for increasing protein lengths and that full structure space enumerations are not feasible within side chain models already for short lengths.

Therefore, we introduced an efficient chain-growth heuristics to tackle the problem of model fitting in Chapter II. Our LatFit approach enables a fast generation of low distance lattice models for full atom protein structures. The procedure is open to arbitrary lattices and enables even the fit of side chain lattice proteins. Thus, we can lift protein studies from backbone-only models to more sophisticated side chain conformations. This is a fundamental step when targeting hard computational problems like structure prediction where a stepwise reduction of the model's abstractions can lead to better results as we have shown later. So far, we analyzed the protein models produced by our LatFit tool for their refinement potential. We presented a constraint-based optimization scheme that allows for the application of a local search in structure space in order to further refine a given lattice protein model. We exemplified its potential and revealed the high performance of the method needed for such a search procedure. The combination of our fast heuristical LatFit approach with a constraint-based local search refinement scheme has the potential to generate close to optimal fits of full atom protein data. We implemented a web frontend for the LatFit tool that is currently to our knowledge the only available lattice fitting program for side chain lattice models.

The following Chapter III presented approaches to tackle the problem of optimal structure prediction. We were able to introduce the first approach that enables the prediction of optimal structures of 3D side chain lattice proteins within the HP model despite the extremely large structure space. Even for longer sequence lengths up to 50 monomers we can compute optimal structures within seconds. This is possible by an extension of the backbone-only Constraint-based Protein Structure Prediction (CPSP) approach by *Backofen and Will* (2006) to side chain models. Therein, an adaption of the formulated Constraint Satisfaction Problems is required but the remaining efficient approach could be maintained. Based on the new approach, we did the first extensive degeneracy study within 3D side chain HP models. We observed an immense number of optimal structures for the majority of sequences.

Further analyses revealed that this immense degeneracy results from the fact that backbone and polar monomers are not constrained by the energy function. Thus, we have introduced an equivalence relation to group optimal structures. The relation focuses on the hydrophobic parts of the structures that are constrained by the energy function and ignores the remaining parts. We could show that the number of the resulting equivalence classes, called core-degeneracy, is on average orders of magnitude lower than the overall number of optimal structures. The core-degeneracy calculation was only possible due to an extension of the CPSP approach. The constraint-based strategy enables the generation of one representative structure per equivalence class and thus the enumeration of the latter. Once again, we could show the high performance of the method resulting in very low runtimes and its flexibility that enabled its application to both backbone-only and side chain models.

In the rest of the chapter, we focused on the prediction of optimal structures for more sophisticated energy models, namely the 20×20-amino acid potentials introduced by *Miyazawa and Jernigan* (1985, 1996). Here, no efficient constraint programming approach is known and thus local search schemes are applied. A standard approach is the application of Simulated Annealing starting from random conformations. We investigated a new protocol that mimics the stages of real protein folding. Therein, the protein is assumed to undergo a spontaneous hydrophobic collapse that forms an initial hydrophobic core. From there, the protein is successively folding into its native structure. We modeled this phenomenon by first computing optimal structures within the HP model that represent the protein after its hydrophobic collapse. These HP-optimal structures are the starting point for a following local search via Simulated Annealing to identify optimal structures within the MJ-energy model, the targeted native structures. We could show that such a protocol outperforms standard Simulated Annealing protocols in both runtime and result quality.

Given the success of the two-step optimization protocol, we plan the integration of the equivalence class enumeration for HP-optimal structures into the procedure. Representatives of different equivalence classes show a different H-monomer arrangement and thus render a different hydrophobic collapse. Since the number of classes grows much slower than the overall number of HP-optimal structures, their representatives are well placed to serve as starting points when optimizing a more sophisticated energy function. Thus, a randomized prediction of HP optimal class representatives to initialize local search schemes should further improve the already performant protocol.

We further envision the extension of the procedure to side chain models. This becomes possible due to two innovations presented within this thesis. First, we have introduced the first approach to compute HP-optimal structures within 3D side chain HP models. Thus, it can be used within the first step of the protocol. Second, we have introduced a local move scheme for side chain models that enables local search within the structure space of side chain models as summarized next. Their combination can yield an efficient protocol to predict optimal protein structures of low abstraction including side chains that could serve as a springboard for full atom structure prediction.

As already mentioned, such a protocol requires the definition of a move set for local search schemes in side chain models. We have introduced such a definition for k-local move neighborhoods in Chapter IV. It defines the allowed structural changes within the steps of local search that are confined to a consecutive interval of a maximal length k. An efficient constraint-based enumeration strategy was presented that can be easily integrated into any local search protocol. Beside the full enumeration of the whole k-local move neighborhood we can use different enumeration schemes available in the constraint programming framework. Among them is for instance a randomized solution generation that is very useful when implementing local search.

A further advantage of the constraint-based approach beside its efficiency is its flexibility. Due to that it can be applied within arbitrary lattices and for any interval length k. Furthermore, it enables the integration of additional structural constraints that are of interest. For instance, we plan the integration of preferred torsion angles along the backbone and to constrain side chain angles according to observations in real protein structures. Due to the fact that we are working in lattice models, we have to derive the targeted angles from according lattice fits rather than from original full atom data. But since we have already introduced fast and well performing approaches for the latter problem, such a data retrieval is easily obtained.

Chapter V studies the phenomenon of co-translational (CT) folding that describes the structure formation of the elongating protein while it is formed at the ribosome. The existance of CT-folding and its importance for proteins to attain their native structure is known from experimental studies. But the sequence features that drive it and the resulting structural patterns are not well known so far or based on thought experiments. We present the first exhaustive study of CT-folding in the HP model in 2D as well as 3D-lattices. To this end, we introduce a sequence classification scheme that enables the identification of sequences that benefit from CT-folding. Based on that we classify sequences into Global-CT, Kinetic-CT, and Non-CT folders. Global-CT and Kinetic-CT folders can adopt their native structure co-translationally while Non-CT classified sequences do not benefit from CT-folding. The difference between Global-CT and Kinetic-CT folders is the targeted structure that is the global energy minimum or a kinetic fold, respectively. We evaluate the procedure via an explicit simulation of CT-folding and show the expected different folding behaviours of the classes.

Using our classification protocol, we are in the position to generate large sets of sequences that benefit from CT-folding and other sets that do not. This enables a comparative study of sequence and structure features specific to one or the other group. Our study revealed subtle differences both in sequence as well as structure space. We found some features theorized in literature to be wrong and were able to identify new properties that can be associated with CT-folding. For instance we did not found CT conformations to be more compact or enriched in local contacts but observed a more N-terminal core and a more compact N- than C-terminal region.

We extended our study to real protein structures and found them biased to the signatures of

CT-folding identified within the HP model. We find that structural domains described by SCOP classes differ in their propensity for CT folding properties and we highlight the α/β class as a strong candidate for CT folding.

Since we found CT-folding to be a powerful and important mechanism to drive protein folding, we set up a new protocol to define and identify protein-like sequences within a protein model. The vast abstractions applied to derive a computionally accessible model often hinder the direct transfer of known proteins into the model. Therefore, a model specific definition of protein-like squences is needed. Often only thermodynamic features are applied as requiring a unique energy minimal ground state. We extended this perspective to incorporate kinetic properties into the definition. To this end, folding simulations are used to evaluate the folding time needed to adapt the native structure. Furthermore, we investigated if the sequence can benefit from CT-folding. Based on such a classification protocol we presented a large data analysis in the 3D-cubic HP model. The resulting data sets of protein-like and non-protein-like sequences can serve as a reasonable base for protein folding studies or for the evaluation of structure prediction approaches. It is the first well defined sequence data set in this model that incorporates both thermodynamic as well as kinetic properties. Since all applied tools implemented are freely available and open to arbitrary energy functions and lattices, the generation of similar data sets of protein-like sequences for other models is directly possible.

In the final Chapter VI we shifted the focus from structure to sequence space to study neutral evolution. This concept, postulated by *Kimura* (1968, 1983), describes the evolution of sequences that all perform the same function via the same structural features. Neutral mutations only change a protein's sequence but do not alter its function, i.e. structure. A key concept to study this process is the investigation of neutral networks that describe the accessible parts of sequence space and the possible evolutionary pathways. Many studies are known from literature investigating simple 2D-lattice models usually based on a full sequence and structure space enumeration. Such enumeration-based studies are not applicable in more complex 3D models due to the much larger structure space. Thus, new non-exhaustive approaches have to be found.

We presented such a protocol to study neutral networks in 3D HP-models. The procedure is split into two parts. Within the first, we design sequences that form a certain structure as their unique optimal one. These sequences populate parts of the targeted neutral network. Thus, we apply a network expansion approach to identify the remaining sequences of the neutral network of interest.

The sequence design problem has been shown by *Berman et al.* (2004) to be a hard NP-complete computational problem on its own. Our approach tackles it via a combination of the generation of a set of sequences that have high chance to be part of the neutral network and their evaluation via the fast and efficient CPSP approach for optimal structure prediction. It is the first approach for 3D HP models that enables the design of non-degenerated sequences for a given structure.

Based on our neutral network expansion protocol, we study the properties and relations of neutral networks in the 3D-cubic model and compare them to results known for 2D-square models. We find the networks much smaller in 3D- compared to 2D-models depicting the increased structural flexibility. Similar to known results, we find larger networks decomposed into several independent connected components. But since the networks are not very large, we do not find the split into a single large component and several small ones reported for 2D. When investigating the mutational distances between connected components, we observe that often already two mutations are sufficient to connect all components into one neutral network. Thus, neutral networks for a certain structure are relatively local in sequence space such that certain functions should be associacted with according sequence features.

We further studied the distribution of all neutral networks within sequence space and found them gathered at low mutational distance too, an observation that we have shown to be unexpected. Evolution can therefore be envisioned as a combination of neutral mutations that preserve a given function and the spawn of new functions via the transition to neighbored neutral networks in sequence space. The larger a neutral network the more likely its function is preserved over time. The strong connectivity between neutral networks implies further that the majority of functions can be reached via the proposed scheme.

Finally, we introduce a notion of folds or structural motifs within the HP model that we find well placed to enable further studies of neutral evolution. The defined H-folds focus on the hydrophobic segments of structures and describe their relative arrangements. Thus, they describe the structural parts that are constrained by the HP-energy function. We introduced a method to derive a unique string encoding for H-folds to enable comparative studies. In addition, we have shown how to enumerate optimal H-folds for a given sequence using the equivalence class enumeration strategy that we have presented for optimal structure grouping. Therefore, we are in the situation to study the evolution of H-folds rather than the evolution of single structures that are more unlikely to be completely preserved over time. Furthermore, we envision the study of loop evolution using H-folds since we know from real proteins that most mutations are accumulated in the flexible interconnecting regions of the structure.

To conclude, we have introduced and used a large set of methods to study a wide range of problems associated with protein folding and evolution. All approaches are implemented and freely available to enable further studies based on our work. Thus, we fill the gap of available efficient tools to study lattice proteins that is highlighted by the frequent usage of our web frontends for ad hoc usage. The introduced approaches enable studies in more complex 3D lattices that have been confined so far to fully enumerable 2D models. Even more important is the applicability of most approaches to side chain models, which opens the door for more realistic lattice model based protein studies in near future.

BIBLIOGRAPHY

BIBLIOGRAPHY

Agashe, V. R., M. C. R. Shastry, and J. B. Udgaonkar (1995), Initial hydrophobic collapse in the folding of barstar, *Nature*, *377*, 754–757.

Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts, , and J. D. Watson (1994), *Molecular Biology of the Cell*, 3rd ed., New York: Garland Science.

Albrecht, A. A., A. Skaliotis, and K. Steinhöfel (2008), Stochastic protein folding simulation in the three-dimensional HP-model, *Comput. Biol. Chem.*, *32*(4), 248–255, doi:10.1016/j.compbiolchem.2008.03.004.

Alder, B. J., and T. E. Wainwright (1959), Studies in molecular dynamics. I. general method, *J. Chem. Phys.*, *31*(2), 459 – 467, doi:10.1063/1.1730376.

Alekshun, M. N., and S. B. Levy (2007), Molecular mechanisms of antibacterial multidrug resistance, *Cell*, *128*(6), 1037–1050, doi:10.1016/j.cell.2007.03.004.

Alexandrov, N. (1993), Structural argument for n-terminal initiation of protein folding, *Protein Sci*, *2*(11), 1989–91.

Anfinsen, C. B. (1973), Principles that govern the folding of protein chains, *Science*, *181*(96), 223 – 230, doi:10.1126/science.181.4096.223.

Anfinsen, C. B., E. Haber, M. Sela, and F. H. White (1961), The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain, *PNAS*, *47*(9), 1309–1314.

Angelani, L., and G. Ruocco (2009), Saddles of the energy landscape and folding of model proteins, *EPL (Europhysics Letters)*, *87*(1), 18,002, doi:10.1209/0295-5075/87/18002.

Atwood, T., P. Campbell, and H. Parish (2006), *Oxford Dictionary of Biochemistry and Molecular Biology*, 2nd edition ed., Oxford University Press.

Backofen, R., and S. Will (1998), Structure prediction in an HP-type lattice with an extended alphabet, in *Proc of German Conference on Bioinformatics (GCB'98)*.

Backofen, R., and S. Will (2001), Optimally compact finite sphere packings — hydrophobic cores in the FCC, in *Proc of CPM'01*, *LNCS*, vol. 2089, pp. 257–272, Springer.

Backofen, R., and S. Will (2002), Excluding symmetries in constraint-based search, *Constraints*, *7*(3), 333–349.

Backofen, R., and S. Will (2006), A constraint-based approach to fast and exact structure prediction in three-dimensional protein models, *Constraints*, *11*(1), 5–30, doi:10.1007/s10601-006-6848-8.

Backofen, R., S. Will, and E. Bornberg-Bauer (1999), Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets, *Bioinformatics*, *15*(3), 234–242.

Backofen, R., S. Will, and P. Clote (2000), Algorithmic approach to quantifying the hydrophobic force contribution in protein folding, in *Proceedings of the Pacific Symposium on Biocomputing (PSB 2000)*, vol. 5, pp. 92–103.

Banavar, J. R., M. Cieplak, and A. Maritan (2004), Lattice tube model of proteins, *Phys. Rev. Lett.*, *93*(23), 238,101, doi:10.1103/PhysRevLett.93.238101.

Banavar, J. R., T. X. Hoang, J. H. Maddocks, A. Maritan, C. Poletto, A. Stasiak, and A. Trovato (2007), Structural motifs of biomolecules, *PNAS*, *104*(44), 17,283–17,286, doi:10.1073/pnas.0704594104.

Bartak, R. (2002), Modelling soft constraints: A survey, *Neural Network World*, *12*, 421–431.

Basharov, M. A. (2000), Cotranslational folding of proteins, *Biochemistry (Moscow)*, *65*(12), 1380–1384, doi:10.1023/A:1002800822475.

Bastolla, U., M. H. Vendruscolo, and M. H. E. Roman (2000), Structurally constrained protein evolution: results from a lattice simulation, *The European Physical Journal B*, *15*(2), 385–397, doi:10.1007/s100510051140.

Bayardo Jr., R. J., and J. D. Pehoushek (2000), Counting models using connected components, in *Proc. of the 7th Nat'l Conf. on AI*.

Beckman Institute at UIUC (2008), Longest ever protein folding movie (may 2008), biophysics Group, University of Illinois at Urbana-Champaign [Online; accessed 19-October-2010].

Berenboym, I., and M. Avigal (2008), Genetic algorithms with local search optimization for protein structure prediction problem, in *Proc. of GECCO '08*, pp. 1097–1098, doi:10.1145/1389095.1389296.

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

Berman, H., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. Shindyalov, and P. Bourne (2000), The Protein Data Bank, *Nucl. Acids Res.*, *28*(1), 235–242, doi:10.1093/nar/28.1.235.

Berman, P., B. DasGupta, D. Mubayi, R. Sloan, G. Turán, and Y. Zhang (2004), The protein sequence design problem in canonical model on 2D and 3D lattices, in *Proc. of Combinatorial Pattern Matching*, vol. 3109, pp. 244–253, Springer.

Berman, P., B. DasGupta, D. Mubayi, R. Sloan, G. Turan, and Y. Zhang (2007), The inverse protein folding problem on 2d and 3d lattices, *Discrete Applied Mathematics*, *155*(6-7), 719–732, doi:10.1016/j.dam.2005.09.018.

Berrera, M., H. Molinari, and F. Fogolari (2003), Amino acid empirical contact energy definitions for fold recognition in the space of contact maps, *BMC Bioinformatics*, *4*(1), 8, doi:10.1186/1471-2105-4-8.

Betancourt, M. R. (2005), Efficient monte carlo trial moves for polypeptide simulations., *J Chem Phys*, *123*(17).

Blackburne, B. P., and J. D. Hirst (2003), Three-dimensional functional model proteins: Structure function and evolution, *J. Chem. Phys.*, *119*, 3453–3461, doi:10.1063/1.1590310.

Blackburne, B. P., and J. D. Hirst (2005), Population dynamics simulations of functional model proteins, *J Chem Phys*, *123*, 154,907–9.

Bloom, J. D., A. Raval, and C. O. Wilke (2007), Thermodynamics of neutral protein evolution, *Genetics*, *175*(1), 255–266, doi:10.1534/genetics.106.061754.

Bock, A., K. Forchhammer, J. Heider, W. Leinfelder, G. Sawers, B. Veprek, and F. Zinoni (1991), Selenocysteine: the 21st amino acid, *Mol Microbiol*, *5*(3), 515–20.

Böckenhauer, H.-J., A. Z. Dayem Ullah, L. Kapsokalivas, and K. Steinhöfel (2008), A local move set for protein folding in triangular lattice models, in *Proc. of WABI '08*, LNBI, pp. 369–381, doi:10.1007/978-3-540-87361-7_31.

Bornberg-Bauer, E. (1997a), Chain growth algorithms for HP-type lattice proteins, in *Proceedings of RECOMB'97*, pp. 47–55.

Bornberg-Bauer, E. (1997b), How are model protein structures distributed in sequence space?, *Biophys J*, *73*(5), 2393–2403.

Bornberg-Bauer, E. (1997c), Simple folding model for hp lattice proteins, in *Bioinformatics, Lecture Notes in Computer Science*, vol. 1278, pp. 125–136, Springer, doi:10.1007/BFb0033211.

Bornberg-Bauer, E. (2002), Randomness, structural uniqueness, modularity and neutral evolution in sequence space of model proteins, *Z. Phys. Chem.*, *216*, 139 – 154, doi: 10.1524/zpch.2002.216.2.139.

Bornberg-Bauer, E., and H. S. Chan (1999), Modeling evolutionary landscapes: mutational stability, topology, and superfunnels in sequence space, *Proc. Natl. Acad. Sci.*, *96*(19), 10,689–10,694.

Bromberg, S., and K. A. Dill (1994), Side-chain entropy and packing in proteins, *Protein Sci*, *3*(7), 997–1009.

Cabrita, L., C. Dobson, and J. Christodoulou (2010), Protein folding on the ribosome, *Curr Opin Struct Biol.*, *20*(1), 33–45.

CASP (2010), Protein Structure Prediction Center, [Online; accessed 08-September-2010], Sponsored by the US National Library of Medicine (NIH/NLM).

CASP8 (2009), Casp8 measures, [Online; accessed 08-September-2010].

Chan, H. S., and E. Bornberg-Bauer (2002), Perspectives on protein evolution from simple exact models, *Applied Bioinformatics*, *1*, 121–144.

Chan, H. S., and K. A. Dill (1996), Comparing folding codes for proteins and polymers, *Proteins-Structure Function and Bioinformatics*, *24*(3), 335–344.

Chikenji, G., Y. Fujitsuka, and S. Takada (2006), Shaping up the protein folding funnel by local interaction: lesson from a structure prediction study, *Proc Natl Acad Sci USA*, *103*(9), 3141–6.

Choi, Y., and C. M. Deane (2010), FREAD revisited: Accurate loop structure prediction using a database search algorithm, *Proteins*, *78*(6), 1431–1440.

Chothia, C. (1992), One thousand families for the molecular biologist., *Nature*, *357*(6379), 543–544.

Chugha, P., and T. G. Oas (2007), Backbone dynamics of the monomeric lambda repressor denatured state ensemble under nondenaturing conditions, *Biochemistry*, *46*(5), 1141–1151, doi:10.1021/bi061371g.

Cipriano, R., L. Gaspero, and A. Dovier (2009), A hybrid solver for large neighborhood search: Mixing Gecode and EasyLocal++, in *Proc. of HM'09*, pp. 141–155, Springer-Verlag, Berlin, Heidelberg, doi:http://dx.doi.org/10.1007/978-3-642-04918-7_11.

Citossi, M., and G. Guigliarelli (2005), Lattice protein models: A computatinoal approach to folding and aggregation phenomena, in *Frontiers of Fundamental Physics*, vol. IV, pp. 355–358, Springer Netherlands, doi:10.1007/1-4020-4339-2.

Clark, P. L. (2004), Protein folding in the cell: reshaping the folding funnel., *Trends Biochem Sci*, *29*(10), 527–534, doi:10.1016/j.tibs.2004.08.008.

Clote, P., and R. Backofen (2000), *Computational Molecular Biology: An Introduction*, Mathematical and Computational Biology, Jon Wiley & Sons, Chichester.

Contreras Martinez, L. M., M. F. J. Veracoechea, P. Pohkarel, A. D. Stroock, F. A. Escobedo, and M. P. Delisa (2006), Protein translocation through a tunnel induces changes in folding kinetics: a lattice model study, *Biotechnol Bioeng*, *94*(1), 105–17.

Covell, D. G., and R. L. Jernigan (1990), Conformations of folded proteins in restricted spaces, *Biochemistry*, *29*(13), 3287–3294.

Crescenzi, P., D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis (1998), On the complexity of protein folding, *Journal of Computational Biology*, *5*(3), 423–65.

Crippen, G. M. (1991), Prediction of protein folding from amino acid sequence over discrete conformation spaces, *Biochemistry*, *30*(17), 4232–4237, doi:10.1021/bi00231a018.

Crippen, G. M., and M. Chhajer (2002), Lattice models of protein folding permitting disordered native states, *J. Chem. Phys.*, *116*, 2261.

Cui, Y., W. H. Wong, E. Bornberg-Bauer, and H. S. Chan (2002), Recombinatoric exploration of novel folded structures: A heteropolymer-based model of protein evolutionary landscapes, *Proc Natl Acad Sci.*, *99*(2), 809–814.

Dal Palu, A., A. Dovier, and F. Fogolari (2004), Constraint Logic Programming approach to protein structure prediction, *BMC Bioinformatics*, *5*, 186.

Dal Palù, A., S. Will, R. Backofen, and A. Dovier (2004), Constraint based protein structure prediction exploiting secondary structure information, in *Proc of CILC'04*, pp. 16–17.

Dal Palù, A., A. Dovier, F. Fogolari, and E. Pontelli (2010), CLP-based protein fragment assembly, *Theory and Practice of Logic Programming*, *10*, 709–724, doi:10.1017/S1471068410000372.

Deane, C. M., M. Dong, F. P. Huard, B. K. Lance, and G. R. Wood (2007), Cotranslational protein folding–fact or fiction?, *Bioinformatics*, *23*(13), i142–148.

Dechter, R., and R. Mateescu (2004), The impact of AND/OR search spaces on constraint satisfaction and counting, in *CP'2004*.

Dill, K. A., S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan (1995), Principles of protein folding – a perspective of simple exact models, *Protein Science*, *4*, 561–602.

Dill, K. A., S. B. Ozkan, M. S. Shell, and T. R. Weikl (2008), The protein folding problem, *Annual Review of Biophysics*, *37*(1), 289–316, doi:10.1146/annurev.biophys.37.092707.153558.

Dima, R., and D. Thirumalai (2002), Exploring protein aggregation and self-propagation using lattice models: Phase diagram and kinetic, *Prot. Sci.*, *11*(5), 1036–1049.

Ding, F., and N. V. Dokholyan (2002), Simple but predictive protein models, *Trends in biotechnology*, *23*(9), 450 – 455, doi:10.1016/j.tibtech.2005.07.001.

Dotu, I., M. Cebrián, P. V. Hentenryck, and P. Clote (2008), Protein structure prediction with large neighborhood constraint programming search, in *Proc of CP'08*, *LNCS*, vol. 5202, pp. 82–96, Springer.

Dyson, H. J. J., P. E. E. Wright, and H. A. A. Scheraga (2006), The role of hydrophobic interactions in initiation and propagation of protein folding., *Proc Natl Acad Sci USA*, doi: 10.1073/pnas.0605504103.

Ekonomiuk, D., M. Kielbasinski, and A. Kolinski (2005), Protein modeling with reduced representation: statistical potentials and protein folding mechanism, *Acta Biochim Pol.*, *52*(4), 741–748.

Ellis, J., F. Huard, C. Deane, S. Srivastava, and G. Wood (2010), Directionality in protein fold prediction., *BMC Bioinformatics*, *11*(1), 172+, doi:10.1186/1471-2105-11-172.

England, J. L., and E. I. Shakhnovich (2003), Structural determinant of protein designability, *Phys. Rev. Lett.*, *90*(21), 218,101, doi:10.1103/PhysRevLett.90.218101.

Eyrich, V. A., D. M. Standley, and R. A. Friesner (1999), Prediction of protein tertiary structure to low resolution: performance for a large and structurally diverse test set, *J Mol Biol*, *288*(4), 725–742, doi:10.1006/jmbi.1999.2702.

Fedorov, A. N., and T. O. Baldwin (1997), Cotranslational protein folding, *J Biol Chem*, *272*(52), 32,715–8.

Feig, M., P. Rotkiewicz, A. Kolinski, J. Skolnick, and C. L. Brooks (2000), Accurate reconstruction of all-atom protein representations from side-chain-based low-resolution models, *Proteins*, *41*(1), 86–97.

Finkelstein, A. V., and A. Y. Badretdinov (1997), Rate of protein folding near the point of thermodynamic equilibrium between the coil and the most stable chain fold, *Folding and Design*, *2*(2), 115 – 121, doi:DOI:10.1016/S1359-0278(97)00016-3.

Finney, J. L. (1975), Volume occupation, environment and accessibility in proteins. the problem of the protein surface, *Journal of Molecular Biology*, *96*(4), 721 – 732, doi:DOI:10.1016/0022-2836(75)90148-5.

Frydman, J., E. Nimmesgern, K. Ohtsuka, and F. U. Hartl (1994), Folding of nascent polypeptide chains in a high molecular mass assembly with molecular chaperones., *Nature*, *370*(6485), 111–117, doi:10.1038/370111a0.

Gan, X., L. Kapsokalivas, A. A. Albrecht, and K. Steinhöfel (2008), A symmetry-free subspace for ab initio protein folding simulations, in *Proc. of BIRD'08*, *CCIS*, vol. 13, pp. 128–139, Springer.

Gasteiger E., H. C. I. I. A. R., Gattiker A., and B. A. (2003), ExPASy: the proteomics server for in-depth protein knowledge and analysis, *Nucleic Acids Res*, *31*(13), 3784 – 3788, doi:10.1093/nar/gkg563.

Gecode (2007), Gecode: Generic constraint development environment, available as an open-source library from `www.gecode.org`.

Geman, S., and D. Geman (1984), Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *PAMI-6*(6), 721–741, doi:10.1109/TPAMI.1984.4767596.

Godzik, A., A. Kolinski, and J. Skolnick (1993), Lattice representations of globular proteins: How good are they?, *J Comp. Chem.*, *14*(10), 1194–1202.

Govindarajan, S., and R. A. Goldstein (1998), On the thermodynamic hypothesis of protein folding, *Proc Natl Acad Sci USA*, *95*(10), 5545–9.

Grüner, W., R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster (1996a), Analysis of rna sequence structure maps by exhaustive enumeration i. neutral networks, *Monatshefte für Chemie*, *127*(4), 355–374, doi:10.1007/BF00810881.

Grüner, W., R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster (1996b), Analysis of rna sequence structure maps by exhaustive enumeration ii. structures of neutral networks and shape space covering, *Monatshefte für Chemie*, *127*(4), 375–389, doi:10.1007/BF00810882.

Guo, Y. Z., E. M. Feng, and Y. Wang (2007), Optimal HP configurations of proteins by combining local search with elastic net algorithm, *J Biochem Biophys Methods*, *70*(3), 335–40.

Gupta, A., J. Manuch, and L. Stacho (2005), Structure-approximating inverse protein folding problem in the 2d hp model, *Journal of Computational Biology*, *12*(10), 1328–1345, doi:10.1089/cmb.2005.12.1328.

Hajek, B. (1988), Cooling schedules for optimal annealing, *Math. Oper. Res.*, *13*(2), 311–329, doi:10.1287/moor.13.2.311.

Hamlin, J., and I. Zabin (1972), Beta-galactosidase: immunological activity of ribosome-bound, growing polypeptide chains, *Proc Natl Acad Sci USA*, *69*(2), 412–6.

Hao, M. H., and H. A. Scheraga (1996), How optimization of potential functions affects protein folding, *Proceedings of the National Academy of Sciences*, *93*(10), 4984–4989.

Hardesty, B., and G. Kramer (2001), Folding of a nascent peptide on the ribosome., *Progress in nucleic acid research and molecular biology*, *66*, 41–66.

Hart, W., and A. Newman (2006), *Handbook of Molecular Biology*, chap. Protein structure prediction with lattice models, pp. 1–24, Chapman & Hall/CRC Computer and Information Science Series, CRC Press, New York.

Hart, W. E. (1997), On the computational complexity of sequence design problems, in *RECOMB*, pp. 128–136.

Hart, W. E., and S. C. Istrail (1996), Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal, *Journal of Computational Biology*, *3*(1), 53–96.

Hart, W. E., and S. C. Istrail (1997), Lattice and off-lattice side chain models of protein folding: linear time structure prediction better than 86% of optimal, *Journal of Computational Biology*, *4*(3), 241–59.

Harvey, W. D., and M. L. Ginsberg (1995), Limited discrepancy search, in *Proceedings of IJCAI'95*, pp. 607–613.

Hastings, W. K. (1970), Monte carlo sampling methods using markov chains and their applications, *Biometrika*, *57*(1), 97–109, doi:10.1093/biomet/57.1.97.

Helling, R., H. Li, R. Melin, J. Miller, N. Wingreen, C. Zeng, and C. Tang (2001), The designability of protein structures, *J Mol Graph Model*, *19*(1), 157–67.

Herráez, A. (2006), Biomolecules in the computer: Jmol to the rescue, *Biochem. Educ*, *34*(4), 255–261.

Heun, V. (2003), Approximate protein folding in the HP side chain model on extended cubic lattices, *Discrete Appl. Math.*, *127*(1), 163–177.

Hinds, D. A., and M. Levitt (1992), A lattice model for protein structure prediction at low resolution, *Proc Natl Acad Sci USA*, *89*(7), 2536–2540.

Hoang, T. X., A. Trovato, F. Seno, J. R. Banavar, and A. Maritan (2004), Geometry and symmetry presculpt the free-energy landscape of proteins, *Proc Natl Acad Sci USA*, *101*(21), 7960–7964, doi:10.1073/pnas.0402525101.

Holley, L. H., and M. Karplus (1989), Protein secondary structure prediction with a neural network., *Proc Natl Acad Sci USA*, *86*(1), 152–156.

Holm, L., C. Ouzounis, C. Sander, G. Tuparev, and G. Vriend (1992), A database of protein structure families with common folding motifs, *Protein Science*, *1*(12), 16911698, doi:10.1002/pro.5560011217.

Hoque, T., M. Chetty, and A. Sattar (2009), Extended hp model for protein structure prediction, *Journal of Computational Biology*, *16*(1), 85–103, doi:10.1089/cmb.2008.0082.

Huard, F. P. E., C. M. Deane, and G. R. Wood (2006), Modelling sequential protein folding under kinetic control, *Bioinformatics*, *22*(14), e203–210.

Humphrey, W., A. Dalke, and K. Schulten (1996), VMD – Visual Molecular Dynamics, *Journal of Molecular Graphics*, *14*, 33–38.

Irbäck, A., and E. Sandelin (2000), On hydrophobicity correlations in protein chains, *Biophys J*, *79*(5), 2252–2258.

Istrail, S., and F. Lam (2009), Combinatorial algorithms for protein folding in lattice models: A survey of mathematical results, *Commun. Inf. Syst.*, *9*(4), 303–346.

Jacob, E., and R. Unger (2007), A tale of two tails: Why are terminal residues of proteins exposed?, *Bioinformatics*, *23*(2), e225–230.

Jacob, E., A. Horovitz, and R. Unger (2007), Different mechanistic requirements for prokaryotic and eukaryotic chaperonins: a lattice study, *Bioinformatics*, *23*(13), i240–i248.

Jefferys, B., L. Kelley, and M. J. E. Sternberg (2010), Protein folding requires crowd control in a simulated cell, *Journal of Molecular Biology*, *397*(5), 1329–1338, doi:10.1016/j.jmb.2010.01.074.

Kabsch, W. (1976), A solution for the best rotation to relate two sets of vectors, *Acta Crystallographica*, *A32*, 922–923.

Kabsch, W. (1978), A discussion of the solution for the best rotation to relate two sets of vectors, *Acta Crystallographica*, *A34*, 827–828.

Kapsokalivas, L., X. Gan, A. A. Albrecht, and K. Steinhöfel (2009), Population-based local search for protein folding simulation in the MJ energy model and cubic lattices, *Comput. Biol. Chem.*, *33*(4), 283–94, doi:10.1016/j.compbiolchem.2009.06.006.

Karlin, S., B. Blaisdell, and P. Bucher (1992), Quantile distributions of amino acid usage in protein classes, *Protein Engineering*, *5*(8), 729–738, doi:10.1093/protein/5.8.729.

Karplus, M. (1997), The Levinthal paradox: yesterday and today., *Folding & design*, *2*(4).

Karplus, M., and J. Kuriyan (2005), Molecular dynamics and protein function, *PNAS*, *102*(19), 6679–6685, doi:10.1073/pnas.0408930102.

Kendrew, J. C., G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips (1953), A three-dimensional model of the myoglobin molecule obtained by X-ray analysis, *Nature*, *181*, 662–666, doi:10.1038/181662a0.

Khodabakhshi, A. H., J. Manuch, A. Rafiey, and A. Gupta (2009), Stable structure-approximating inverse protein folding in 2D hydrophobic-polar-cysteine (HPC) model, *Journal of Computational Biology*, *16*(1), 19–30, doi:10.1089/cmb.2008.0096.

Kimura, M. (1968), Evolutionary rate at the molecular level, *Nature*, *217*, 624 – 626, doi:10.1038/217624a0.

Kimura, M. (1983), *The neutral theory of molecular evolution*, Cambridge University Press.

Kimura, M. (1991), The neutral theory of molecular evolution: A review of recent evidence, *Jpn J Genet.*, *66*, 367 – 386, doi:10.1266/jjg.66.367.

Kirkpatrick, S., J. C. D. Gelatt, and M. P. Vecchi (1983), Optimization by simulated annealing, *Science*, *220*(4598), 671–680.

Klemm, K., C. Flamm, and P. F. Stadler (2008), Funnels in energy landscapes, *The European Physical Journal B*, *63*, 387–391, doi:10.1140/epjb/e2008-00136-7.

Kmiecik, S., and A. Kolinski (2007), Characterization of protein-folding pathways by reduced-space modeling, *PNAS*, *104*(30), 12,330–12,335.

Koehl, P., and M. Delarue (1998), Building protein lattice models using self-consistent mean field theory, *J. Chem. Phys.*, *108*, 9540–9549, doi:10.1063/1.476402.

Koh, S. K., G. K. Ananthasuresh, and S. Vishveshwara (2005), A deterministic optimization approach to protein sequence design using continuous models, *The International Journal of Robotics Research*, *24*, 109–130.

Kolb, V. A. (2001), Cotranslational protein folding, *Mol Biol (Mosk)*, *35*(4), 682–90.

Kolb, V. A., E. V. Makeyev, and A. S. Spirin (2000), Co-translational folding of an eukaryotic multidomain protein in a prokaryotic translation system, *J Biol Chem*, *275*(22), 16,597–601.

Kolinski, A., and J. Skolnick (2004), Reduced models of proteins and their applications, *Polymer*, *45*(2), 511 – 524, doi:10.1016/j.polymer.2003.10.064.

Kramer, G., D. Boehringer, N. Ban, and B. Bukau (2009), The ribosome as a platform for co-translational processing, folding and targeting of newly synthesized proteins, *Nat Struct Mol Biol*, *16*, 589–597.

Krasnogor, N., B. Blackburne, E. Burke, and J. Hirst (2002), Multimeme algorithms for protein structure prediction, in *Proc. of Parallel Problem Solving from Nature PPSN VII*, *LNCS*, vol. 2439, pp. 769–778, doi:10.1007/3-540-45712-7_74.

Kryshtafovych, A., O. Krysko, P. Daniluk, Z. Dmytriv, and K. Fidelis (2009), Protein structure prediction center in CASP8, *Proteins: Structure, Function, and Bioinformatics*, *77*, 5–9, doi:10.1002/prot.22517.

Laio, A., and C. Micheletti (2006), Are structural biases at protein termini a signature of vectorial folding?, *Proteins*, *62*(1), 17–23.

Lau, K. F., and K. A. Dill (1989), A lattice statistical mechanics model of the conformational and sequence spaces of proteins, *Macromol.*, *22*(10), 3986–3997.

Laurèn, J., D. A. Gimbel, H. B. Nygaard, J. W. Gilbert, and S. M. Strittmatter (2009), Cellular prion protein mediates impairment of synaptic plasticity by amyloid-β oligomers, *Nature*, *457*, 1128–1132, doi:10.1038/nature07761.

Lesh, N., M. Mitzenmacher, and S. Whitesides (2003), A complete and effective move set for simplified protein folding, in *Proceedings of the seventh annual international conference on Research in computational molecular biology (RECOMB'03)*, pp. 188–195.

Levinthal, C. (1968), Are there pathways for protein folding?, *Extrait du Journal de Chimie Physique*, *65*(1).

Li, H., R. Helling, C. Tang, and N. Wingreen (1996), Emergence of preferred structures in a simple model of protein folding, *Science*, *273*(5275), 666–669, doi:10.1126/science.273.5275.666.

Li, H., C. Tang, and N. S. Wingreen (2002), Designability of protein structures: a lattice-model study using the Miyazawa-Jernigan matrix, *Proteins*, *49*, 403–412.

Li, X. (2007), Protein folding based on simulated annealing algorithm, in *Proc. of ICNC'07, Natural Computations*, vol. 4, pp. 256 – 259, doi:10.1109/ICNC.2007.583.

Madras, N., and A. D. Sokal (1987), Nonergodicity of local, length-conserving Monte Carlo algorithms for the self-avoiding walk, *Journal of Statistical Physics*, *47*(3-4), 573–595.

Madras, N., and A. D. Sokal (1988), The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk, *Journal of Statistical Physics*, *50*(1-2), 109–186.

Mann, M. (2006), Vollstaendige aufzählung der optimalen strukturen von gitterproteinen durch dynamische zerlegung des assoziierten constraint satisfaction problems, Diplomarbeit, Friedrich Schiller University Jena.

Mann, M., and A. D. Palu (2010), Lattice model refinement of protein structures, in *Proc of WCB'10*, p. 7, arXiv:1005.1853.

Mann, M., G. Tack, and S. Will (2007a), Decomposition during search for propagation-based constraint solvers, *Tech. rep.*, ALU Freiburg.

Mann, M., S. Will, and R. Backofen (2007b), The energy landscape library - a platform for generic algorithms, in *BIRD'07 - 1st international Conference on Bioinformatics Research and Development*, vol. 217, pp. 83–86, Oesterreichische Computer Gesellschaft.

Mann, M., D. Maticzka, R. Saunders, and R. Backofen (2008a), Classifying protein-like sequences in arbitrary lattice protein models using LatPack, *HFSP Journal*, *2*(6), 396–404, doi:10.2976/1.3027681, special issue on protein folding: experimental and theoretical approaches.

Mann, M., S. Will, and R. Backofen (2008b), CPSP-tools - exact and complete algorithms for high-throughput 3D lattice protein studies, *BMC Bioinformatics*, *9*, 230, doi:10.1186/1471-2105-9-230.

Mann, M., R. Backofen, and S. Will (2009a), Equivalence classes of optimal structures in HP protein models including side chains, in *Proceedings of the Fifth Workshop on Constraint Based Methods for Bioinformatics (WCB09)*.

Mann, M., M. A. Hamra, K. Steinhöfel, and R. Backofen (2009b), Constraint-based local move definitions for lattice protein models including side chains, in *Proceedings of the Fifth Workshop on Constraint Based Methods for Bioinformatics (WCB09)*, arXiv:0910.3880.

Mann, M., C. Smith, M. Rabbath, M. Edwards, S. Will, and R. Backofen (2009c), CPSP-web-tool : a server for 3D lattice protein studies, *Bioinformatics*, *25*(5), 676–677, doi:10.1093/bioinformatics/btp034.

Maňuch, J., and D. R. Gaur (2008), Fitting protein chains to cubic lattice is NP-complete, *Journal of bioinformatics and computational biology*, *6*(1), 93–106.

Mazzoni, L. N., and L. Casetti (2006), Curvature of the energy landscape and folding of model proteins, *Physical Review Letters*, *97*(21), 218,104.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equation of state calculations by fast computing machines, *The Journal of Chemical Physics*, *21*(6), 1087–1092.

Miao, J., J. Kleinseetharaman, and H. Meirovitch (2004), The optimal fraction of hydrophobic residues required to ensure protein collapse, *J Mol. Bio.*, *344*(3), 797–811, doi:http://dx.doi.org/10.1016/j.jmb.2004.09.061.

Mirny, L., and E. Shakhnovich (2001), Protein folding theory: From lattice to all-atom models, *Annual Review of Biophysics and Biomolecular Structure*, *30*(1), 361–396, doi:10.1146/annurev.biophys.30.1.361.

Miyazawa, S., and R. L. Jernigan (1985), Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation, *Macromolecules*, *18*(3), 534–552, doi:10.1021/ma00145a039.

Miyazawa, S., and R. L. Jernigan (1996), Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading, *J Mol Biol*, *256*(3), 623–44.

Mizuguchi, K., C. M. Deane, T. L. Blundell, M. S. Johnson, and J. P. Overington (1998), Joy: protein sequence-structure representation and analysis, *Bioinformatics*, *14*(7), 617–23.

Morrissey, M. P., Z. Ahmed, and E. I. Shakhnovich (2004), The role of cotranslation in protein folding: a lattice model study, *Polymer*, *45*(2), 557–571.

Munson, M., S. Balasubramanian, K. Fleming, A. Nagi, R. O'Brien, J. Sturtevant, and L. Regan (1996), What makes a protein a protein? hydrophobic core designs that specify stability and structural properties., *Protein Science*, *5*(8), 1584–1593.

Murzin, A. G., S. E. Brenner, T. Hubbard, and C. Chothia (1995), Scop: a structural classification of proteins database for the investigation of sequences and structures, *J Mol Biol*, *247*(4), 536–40.

Nature-Editorial (2010), Oil spills: microorganisms to the rescue?, *Nature Reviews Microbiology*, *8*, 462, doi:10.1038/nrmicro2404.

Newman, A. (2002), A new algorithm for protein folding in the HP model, in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms.*

Ngo, J. T., and J. Marks (1992), Computational complexity of a problem in molecular structure prediction, *Protein Eng.*, *5*(4), 313–321, doi:10.1093/protein/5.4.313.

Nicola, A. V., W. Chen, and A. Helenius (1999), Co-translational folding of an alphavirus capsid protein in the cytosol of living cells, *Nat Cell Biol*, *1*(6), 341–5.

Noirel, J., and T. Simonson (2007), Neutral evolution of protein-protein interactions: a computational study using simple models, *BMC Structural Biology*, *7*, 79, doi:10.1186/1472-6807-7-79.

Noirel, J., and T. Simonson (2008), Neutral evolution of proteins: The superfunnel in sequence space and its relation to mutational robustness, *J. Chem. Phys.*, *129*, 185,104, doi:10.1063/1.2992853.

Nunnally, B. K., and I. S. Krull (Eds.) (2003), *Prions and Mad Cow Disease*, CRC Press.

Pace, C., B. Shirley, M. McNutt, and K. Gajiwala (1996), Forces contributing to the conformational stability of proteins, *FASEB J.*, *10*(1), 75–83.

Palù, A. D., A. Dovier, and E. Pontelli (2005), A new constraint solver for 3D lattices and its application to the protein folding problem, in *Proc. of Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'05)*, pp. 48–63, Springer.

Palù, A. D., A. Dovier, and E. Pontelli (2010), Computing approximate solutions of the protein structure determination problem using global constraints on discrete crystal lattices, *J of Data Mining and Bioinformatics*, *4*(1), 1 – 20.

Park, B., and M. Levitt (1995), The complexity and accuracy of discrete state models of protein structure, *J Mol Biol*, *249*, 493–507.

Pascarella, S., and P. Argos (1992a), A data bank merging related protein structures and sequences, *Protein Engineering*, *5*(2), 121–137, doi:10.1093/protein/5.2.121.

Pascarella, S., and P. Argos (1992b), Analysis of insertions/deletions in protein structures, *Journal of Molecular Biology*, *224*(2), 461 – 471, doi:10.1016/0022-2836(92)91008-D.

Pauling, L., and R. B. Corey (1951a), Atomic coordinates and structure factors for two helical configurations of polypeptide chains, *PNAS*, *37*(11), 729–740.

Pauling, L., and R. B. Corey (1951b), Configurations of polypeptide chains with favored orientations around single bonds: Two new pleated sheets, *PNAS*, *37*(5), 235–240.

Pesant, G., and M. Gendreau (1999), A constraint programming framework for local search methods, *Journal of Heuristics*, *5*(3), 255–279.

Pierri, C., A. D. Grassi, and A. Turi (2008), Lattices for ab initio protein structure prediction, *Proteins*, *73*(2), 351–361, doi:10.1002/prot.22070.

Plimpton, S. (1995), Computational limits of classical molecular dynamics simulations, *Computational Materials Science*, *4*, 361–364, doi:10.1.1.42.9644.

Ponty, Y., R. Istrate, E. Porcelli, and P. Clote (2008), LocalMove: computing on-lattice fits for biopolymers, *Nucleic Acids Res*, *36*(2), W216–W222.

Potzsch, S., G. Scheuermann, M. Wolfinger, C. Flamm, and P. Stadler (2006), Visualization of lattice-based protein folding simulations, in *In Proc. of V'06: Conference on Information Visualization*, pp. 89–94, doi:10.1109/IV.2006.127.

Prusiner, S. B. (1998), Prions, *Proceedings of the National Academy of Sciences of the United States of America*, *95*(23), 13,363–13,383, doi:10.1073/pnas.95.23.13363.

Rabbath, M. (2008), Constraint approach for protein structure prediction in the side chain HP model, Master thesis, Albert Ludwigs University Freiburg.

Ramachandran, G. N., C. Ramakrishnan, and V. Sasisekharan (1963), Stereochemistry of polypeptide chain configurations, *J Mol Biol.*, *7*, 95–99.

Ramnarayan, K., H. Bohr, and K. Jalkanen (2008), Classification of protein fold classes by knot theory and prediction of folds by neural networks: A combined theoretical and experimental approach, *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, *119*, 265–274, doi:10.1007/s00214-007-0285-7.

Régin, J.-C. (1994), A filtering algorithm for constraints of difference in CSPs, in *Proc. of 12th National Conference on AI*, pp. 362–367.

Reidys, C. (2009), Local connectivity of neutral networks, *Bulletin of Mathematical Biology*, *71*(2), 265–290, doi:10.1007/s11538-008-9356-8.

Reidys, C., P. F. Stadler, and P. Schuster (2006), Generic properties of combinatory maps: Neutral networks of rna secondary structures, *Bulletin of Mathematical Biology*, *59*(2), 339–397, doi:10.1007/BF02462007.

Renner, A., and E. Bornberg-Bauer (1997), Exploring the fitness landscapes of lattice proteins, *Pac Symp Biocomput.*, pp. 361–372.

Reva, B. A., D. S. Rykunov, A. J. Olson, and A. V. Finkelstein (1995), Constructing lattice models of protein chains with side groups, *Journal of Computational Biology*, *2*(4), 527–535, doi:10.1089/cmb.1995.2.527.

Reva, B. A., D. S. Rykunov, A. V. Finkelstein, and J. Skolnick (1998), Optimization of protein structure on lattices using a self-consistent field approach, *Journal of Computational Biology*, *5*(3), 531–538, doi:10.1089/cmb.1998.5.531.

Rivière, D., V. Desvignes, E. Pelletier, S. Chaussonnerie, S. Guermazi, J. Weissenbach, T. Lia, P. Camacho, and A. Sghir (2009), Towards the definition of a core of microorganisms involved in anaerobic digestion of sludge, *ISME Journal*, *3*, 700–714, doi:10.1038/ismej.2009.2.

Rose, G. D., P. J. Fleming, J. R. Banavar, and A. Maritan (2006), A backbone-based theory of protein folding, *PNAS*, *103*(45), 16,623 – 16,633, doi:10.1073/pnas.0606843103.

Sali, A., E. Shakhnovich, and M. Karplus (1994), Kinetics of protein folding. A lattice model study of the requirements for folding to the native state, *J Mol Biol*, *235*(5), 1614–1636.

Sanchez, I. E., M. Morillas, E. Zobeley, T. Kiefhaber, and R. Glockshuber (2004), Fast folding of the two-domain semliki forest virus capsid protein explains co-translational proteolytic activity, *J Mol Biol*, *338*(1), 159–67.

Sandelin, E. (2004), On hydrophobicity and conformational specificity in proteins, *Biophys J*, *86*, 23–30.

Saunders, R., and C. M. Deane (2009), Protein structure prediction begins well but ends badly, *Proteins*, *78*(5), 12821290, doi:10.1002/prot.22646.

Saunders, R., and C. M. Deane (2010), Synonymous codon usage influences the local protein structure observed, *Nucl. Acids Res.*, p. gkq495, doi:10.1093/nar/gkq495.

Schuster, P. (2008), Modeling in biological chemistry. from biochemical kinetics to systems biology, *Chemical Monthly*, *139*(4), 427–446, doi:10.1007/s00706-008-0892-6.

Schuster, P., W. Fontana, P. F. Stadler, and I. L. Hofacker (1994), From sequences to shapes and back: a case study in RNA secondary structures, *Proc. Royal Society London B*, *255*(1344), 279–284.

Scott, M. D., and J. Frydman (2003), Aberrant protein folding as the molecular basis of cancer, in *Protein Misfolding and Disease*, *Methods in Molecular Biology*, vol. 232, pp. 67–76, Humana Press, doi:10.1385/1-59259-394-1:67.

Shakhnovich, E. I. (1996), Modeling protein folding: the beauty and power of simplicity, *Fold Des.*, *1*(3), R50–54.

Shakhnovich, E. I., and A. M. Gutin (1993), Engineering of stable and fast-folding sequences of model proteins, *PNAS*, *90*(15), 7195–7199.

Shannon, M. A., P. W. Bohn, M. Elimelech, J. G. Georgiadis, B. J. M. nas, and A. M. Mayes (2008), Science and technology for water purification in the coming decades, *Nature*, *452*, 301–310, doi:10.1038/nature06599.

Shen, M., and A. Sali (2006), Statistical potential for assessment and prediction of protein structures, *Protein Science*, *15*(11), 2507–2524, doi:10.1110/ps.062416606.

Shortle, D., H. S. Chan, and K. A. Dill (1992), Modeling the effects of mutations on the denatured states of proteins., *Prot Sci*, *1*, 201–215.

Sikorski, A., and J. Skolnick (1990), Dynamic Monte Carlo simulations of globular protein folding. model studies of in vivo assembly of four helix bundles and four member beta-barrels, *J Mol Biol*, *215*(1), 183–98.

Simmons, K. J., I. Chopra, and C. W. G. Fishwick (2010), Structure-based discovery of antibacterial drugs, *Nature Reviews Microbiology*, *8*, 501–510, doi:10.1038/nrmicro2349.

Sippl, M. (1990), Calculation of conformational ensembles from potentials of mean force. an approach to the knowledge-based prediction of local structures in globular proteins, *J. Mol. Biol.*, *213*(4), 859–883.

Skolnick, J., and A. Kolinski (1991), Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics, *J Mol Biol*, *221*(2), 499–531.

Smith, A. (2003), Protein misfolding, *Nature*, *426*(6968 (Insight)), 883–909, special Insight issue on protein misfolding edited by A. Smith.

Song, J., J. Cheng, and T. Zheng (2006), Protein 3D HP model folding simulation based on ACO, in *Proc. of ISDA'06*, vol. 1, pp. 410–415, doi:10.1109/ISDA.2006.221.

Steinhöfel, K., A. Skaliotis, and A. A. Albrecht (2007), Stochastic protein folding simulation in the d-dimensional HP-model, in *Proceedings of the 1st Conference on BioInformatics Research and Development*, pp. 381–394, Springer.

Sun, Z., X. Xia, Q. Guo, and D. Xu (1999), Protein structure prediction in a 210-type lattice model: Parameter optimization in the genetic algorithm using orthogonal array, *Journal of Protein Chemistry*, *18*, 39–46, doi:10.1023/A:1020643331894.

Swetnam, A. D., and M. P. Allen (2010), Improving the wanglandau algorithm for polymers and proteins, *J Comp Chem*, doi:10.1002/jcc.21660, early View (Articles online in advance of print).

Sykes, M. F., A. J. Guttmann, M. G. Watts, and P. D. Roberts (1972), The asymptotic behaviour of selfavoiding walks and returns on a lattice, *Journal of Physics A: General Physics*, *5*(5), 653, doi:10.1088/0305-4470/5/5/006.

Tacker, M., P. Stadler, E. Bornberg-Bauer, I. Hofacker, and P. Schuster (1996), Algorithmic independent properties of RNA secondary structure predictions, *Eur. Biophys. J.*, *25*, 115–130.

Taylor, W. R. (2006), Topological accessibility shows a distinct asymmetry in the folds of $\beta\alpha$ proteins, *FEBS Letters*, *580*(22), 5263–5267, doi:10.1016/j.febslet.2006.08.070.

Thachuk, C., A. Shmygelska, and H. H. Hoos (2007), A replica exchange Monte Carlo algorithm for protein folding in the HP model, *BMC Bioinformatics*, *8*, 342.

Thanaraj, T. A., and P. Argos (1996), Ribosome-mediated translational pause and protein domain organization, *Protein Sci*, *5*(8), 1594–612.

Theurillat, J.-P., B. Dreier, G. Nagy-Davidescu, B. Seifert, S. Behnke, U. Zürrer-Härdi, F. Ingold, A. Plückthun, and H. Moch (2010), Designed ankyrin repeat proteins: a novel tool for testing epidermal growth factor receptor 2 expression in breast cancer, *Modern Pathology*, doi:10.1038/modpathol.2010.103.

Tobi, D., and R. Elber (2000), Distance-dependent, pair potential for protein folding: Results from linear optimization, *Proteins*, *41*, 40–46.

Tokuriki, N., F. Stricher, L. Serrano, and D. S. Tawfik (2008), How protein stability and new functions trade off, *PLoS Comput Biol.*, *4*(2), e1000,002, doi:10.1371/journal.pcbi.1000002.

Tsang, E. (1993), *Foundations of Constraint Satisfaction*, Academic Press.

Tyers, M., and M. Mann (2003), From genomics to proteomics, *Nature*, *422*, 193–197, doi:10.1038/nature01510.

Ullah, A. D., and K. Steinhöfel (2010), A hybrid approach to protein folding problem integrating constraint programming with local search, *BMC Bioinformatics*, *11*(Suppl 1), S39, doi:10.1186/1471-2105-11-S1-S39.

Ullah, A. D., L. Kapsokalivas, M. Mann, and K. Steinhöfel (2009), Protein folding simulation by two-stage optimization, in *Proc. of ISICA'09*, *CCIS*, vol. 51, pp. 138–145, Springer, Wuhan, China, doi:10.1007/978-3-642-04962-0_16.

Unger, R., and J. Moult (1993), Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications, *Bull Math Biol*, *55*(6), 1183–98, doi:10.1007/BF02460703.

van Nimwegen, E., J. P. Crutchfield, and M. Huynen (1999), Neutral evolution of mutational robustness, *PNAS*, *96*(17), 9716–9720, doi:10.1073/pnas.96.17.9716.

Černý, V. (1985), Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *J of Optimization Theory and Applications*, *45*(1), 41–51, doi:10.1007/BF00940812.

Wagner, A. (2008), Neutralism and selectionism: a network-based reconciliation, *Nature Reviews Genetics*, *9*, 965 – 974, doi:10.1038/nrg2473.

Wang, G., and R. L. Dunbrack (2003), PISCES: a protein sequence culling server, *Bioinformatics*, *19*(12), 1589–91.

Wang, G., and R. L. Dunbrack (2005), PISCES: recent improvements to a PDB sequence culling server, *Nucleic Acids Res*, *33*(Web Server issue), W94–8.

Wang, P., and D. K. Klimov (2007), Lattice simulations of cotranslational folding of single domain proteins, *Proteins*, *70*(3), 925–937.

Wang, T., J. Miller, N. S. Wingreen, C. Tang, and K. A. Dill (2000), Symmetry and designability for lattice protein models, *The Journal of Chemical Physics*, *113*(18), 8329–8336, doi:10.1063/1.1315324.

Wiedersich, J., S. Köhler, A. Skerra, and J. Friedrich (2008), Temperature and pressure dependence of protein stability: The engineered fluorescein-binding lipocalin FluA shows an elliptic phase diagram, *PNAS*, *105*(15), 5756–5761, doi:10.1073/pnas.0710409105.

Will, S. (2005), Exact, constraint-based structure prediction in simple protein models, Ph.D. thesis, Friedrich-Schiller-Universität Jena.

Will, S., and M. Mann (2006), Counting protein structures by DFS with dynamic decomposition, in *Proc of WCB'06*, pp. 83–90.

Wittung-Stafshede, P. (2002), Role of cofactors in protein folding, *Acc Chem Res*, *35*(4), 201–208, doi:10.1021/ar010106e.

Wolfinger, M., S. Will, I. Hofacker, R. Backofen, and P. Stadler (2006), Exploring the lower part of discrete polymer model energy landscapes, *Europhysics Letters*, *74*(4), 725–732.

Wolynes, P. G., J. N. Onuchic, and D. Thirumalai (1995), Navigating the folding routes, *Science*, *267*(5204), 1619 – 1620.

Wroe, R., E. Bornberg-Bauer, and H. S. Chan (2005), Comparing folding codes in simple heteropolymer models of protein evolutionary landscape: robustness of the superfunnel paradigm, *Biophys J*, *88*(1), 118–131.

Wroe, R., H. S. Chan, and E. Bornberg-Bauer (2007), A structural model of latent evolutionary potentials underlying neutral networks in proteins, *HFSP Journal*, *1*, 79–87, doi:10.2976/1.2739116.

Wüthrich, K. (1990), Protein structure determination in solution by NMR spectroscopy, *J Biol Chem.*, *265*(36), 22,059–22,062.

Xia, Y., and M. Levitt (2004a), Simulating protein evolution in sequence and structure space., *Curr Opin Struct Biol*, *14*(2), 202–207, doi:http://dx.doi.org/10.1016/j.sbi.2004.03.001.

Xia, Y., and M. Levitt (2004b), Funnel-like organization in sequence space determines the distributions of protein stability and folding rate preferred by evolution., *Proteins*, *55*(1), 107–114, doi:http://dx.doi.org/10.1002/prot.10563.

Ying, B. W., H. Taguchi, and T. Ueda (2006), Co-translational binding of groel to nascent polypeptides is followed by post-translational encapsulation by groes to mediate protein folding, *J Biol Chem*, *281*(31), 21,813–21,819.

Yue, K., and K. A. Dill (1992), Inverse protein folding problem: designing polymer sequences, *Proc Natl Acad Sci.*, *89*(9), 4163–4167.

Yue, K., and K. A. Dill (1995), Forces of tertiary structural organization in globular proteins, *Proc Natl Acad Sci.*, *92*(1), 146–150.

Yue, K., K. M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Shakhnovich, and K. A. Dill (1995), A test of lattice protein folding algorithms, *Proc Natl Acad Sci.*, *92*(1), 325–329.

Zhang, G., M. Hubalewska, and Z. Ignatova (2009), Transient ribosomal attenuation coordinates protein synthesis and co-translational folding, *Nat Struct Mol Biol*, *16*, 274–280.

Zhang, J., S. C. Kou, and J. S. Liu (2007), Biopolymer structure simulation and optimization via fragment regrowth Monte Carlo., *J Chem Phys*, *126*(22), 225,101.