Energy Landscapes of Biopolymers

DISSERTATION
zur Erlangung des akademischen Grades
Doktor rerum naturalium

Vorgelegt der
Fakultät für Chemie
der Universität Wien

von

Mag. Michael Wolfinger

am Institut für Theoretische Chemie und Molekulare
Strukturbiologie

im Oktober 2004

Dank an alle,

die zum Gelingen dieser Arbeit beigetragen haben:

Peter Stadler, Ivo Hofacker, Peter Schuster.

Ingrid Abfalter, Stephan Bernhart, Lukas Endler, Christoph Flamm, Claudia Fried, Kurt Grünberger, Jörg Hackermüller, Ulli Langhammer, Rainer Machne, Ulli Mückstein, Stefan Müller, Sonja Prohaska, Bärbel Stadler, Camille Stephan-Otto Attolini, Roman Stocsits, Andreas Svrcek-Seiler, Andrea Tanzer, Caroline Thurner, Stefan Washietl, Stefanie Widder, Christina Witwer, Judith Ivansits, Judith Jakubetz.

Frieda Wolfinger, Sepp Jürgen Wolfinger.

Zusammenfassung

Biomoleküle wie DNA, RNA oder Proteine bilden die molekulare Basis aller bekannten Lebensformen. Die Möglichkeit von Biopolymeren, in eine wohldefinierte native Struktur zu falten stellt eine Notwendigkeit für biologisch relevante Moleküle dar. Um Biomoleküle einer theoretischen Untersuchung zuführen zu können ist es notwendig, ein gewisses Abstraktionsniveau einzuführen. Im Fall von RNA wird diese Abstraktion durch Sekundärstrukturen erreicht. Basierend auf experimentell gemessenen Energieparametern wurden an unserem Institut in den vergangenen Jahren effiziente Algorithmen zur computergestützten Behandlung von RNA Sekundärstrukturen entwickelt und als `Vienna RNA Package` einer breiten Öffentlichkeit zugänglich gemacht. Proteine werden oft als "self-avoiding walk" auf verschiedenen Gittern als Sequenzen aus zwei Monomertypen, nämlich hypdrophob (**H**) und polar (**P**) dargestellt.
Ein grundlegender Baustein bei der Untersuchung von komplexen molekularen Systemen ist die Untersuchung der Energiefläche, auf der sich die Dynamik des Systems abspielt. Detailliertes Verständnis der strukturellen Eigenschaften von komplexen Landschaften ist daher essenziell für die Biophysik von Heteropolymeren. Struktur-formende Prozesse sowie die Kinetik von Biopolymeren sind inherent verknüpft mit den topologischen Gegebenheiten der Energielandschaft, im Speziellen Basins und Energiebarrieren.
Im Rahmen dieser Dissertation wird ein effizienter Algorithmus zur Untersuchung der Eigenschaften von Energielandschaften, z.B. Anzahl der lokalen Minima oder Verteilung von Basins vorgestellt. Durch hierachische Anordnung der Konformationen ist der Algorithmus in der Lage, die Energielandschaft als so genannten *barrier tree* darzustellen. Barrier trees vermitteln einen Eindruck von der gesamtheitlichen Struktur sowie der Rauhheit der Energielandschaft.
Ein stochastischer Algorithmus zur Simulation der Faltungskinetik von RNA, der auf elementaren Schritten im Konformationsraum basiert, wurde um das Gebiet der Gitterproteine ergänzt. Eine erweiterte Form einer Arrhenius-artigen macrostate-Kinetik, die auf barrier trees formuliert werden kann, wird mit der stochastischen Kinetik verglichen. Der Vorteil der macrostate-Kinetik ist eine drastische Reduktion an Rechenzeit. Dadurch kann die Dynamik von Biomolekülen in der Grösse von tRNA binnen Minuten untersucht werden.
Weiters wird ein neuer Ansatz zur Berechnung des energetisch niedrigen Teils der Energielandschaft von Gitterproteinen präsentiert.

Abstract

Biomolecules like DNA, RNA or proteins form the molecular basis of all known forms of life. The ability of biopolymers to fold into a well-defined native state is a prerequisite for biologically functional molecules. In order to treat biomolecules within a theoretical framework, a reasonable level of abstraction or coarse-graining is needed. RNA can be modeled conveniently by means of secondary structures. Based upon experimentally measured energy parameters, efficient dynamic programming algorithms for a computational treatment of RNA secondary structures have been developed at our institute and made available as the `Vienna RNA Package`. Proteins are often modeled as self-avoiding walks on various lattices with a sequence consisting of only two monomer types, hydrophobic **H** and polar **P** residues.

A fundamental prerequisite in complexity studies of molecular systems is certainly a thorough investigation of the energy surface on which the system dynamics evolve. A detailed understanding of structural features of complex landscapes thus lies at the heart of the biophysics of heteropolymers. Kinetics and structure formation processes of biopolymers are crucially determined by the topological details of the energy landscape, i.e. basins and barriers separating them.

We introduce an efficient algorithm for measurement of features of energy landscapes, such as the number of local minima, the size distribution of basins of attraction or thermodynamic quantities. The algorithm is capable of constructing a hierarchical order of conformations that can be represented compactly in so called *barrier trees*, giving an impression of the shape and ruggedness of the energy landscape.

A stochastic algorithm for the simulation of kinetic folding of RNA, based on elementary steps in conformation has been extended to the field of lattice proteins. We compare results from an extended Arrhenius-type macrostate kinetics, that can be formulated on the barrier tree with results from the stochastic simulation. A major advantage of the coarse-grained dynamics is time efficiency, allowing computational treatment of tRNA size molecules' dynamics within a time-scale of several minutes.

We will further present a novel approach to generate the lowest-energy part of lattice protein energy landscapes based on elementary steps starting from a low-energy state.

Contents

1 Introduction

The last 150 years were certainly of utmost importance for molecular biology. When Charles Darwin proposed his first empirical theory of biological evolution in 1859, he suggested that the diversity and complexity of present day organisms can be explained on the basis of two key principles: inheritable *variation* and natural *selection*. Although the laws and mechanisms of variation had not been accepted in the nineteenth century, his theory became one of the most influential contributions to natural sciences and nowadays forms a cornerstone of a modern view of the basis of life.

However, almost a century should pass after Darwin's famous contributions that scientists elucidated the *molecular* basis of life: Biomolecules or biopolymers, mostly linear polymers consisting of covalently bounded monomers, are the most important ingredients in the cookbook of life. A common principle to all polymeric macromolecules in living systems are highly ordered chemical entities with specific sequences of monomeric subunits that are responsible for specific, discrete structures and functions. There are three fundamental principles that arise within this context [91]:

- function of a biopolymer is determined by its unique structure
- non-covalent interactions play a critical role in biopolymer structure and function
- the specific sequence that is built from monomeric subunits encodes information that is crucial for all living elements

Most biopolymers are heteropolymers, which means that their sequence is built from a handful of different monomers. DNA and RNA, for example, are made of four different nucleotides, whereas 20 different amino acids form proteins.

As the structure is responsible for a biopolymer's function, it seems fair to say that a theoretical chemist's main interest lies in the three dimensional shape of a biomolecule. As a matter of fact, it would be desirable to calculate a biopolymer's native structure only with knowledge of its sequence. However, the huge number of atoms and the immense dimensionality of conformation space make such calculations impossible with present day computer resources. It is thus necessary to shift the problem towards a direction that is computationally feasible, i.e. it is necessary to coarse-grain the problem. Within the framework of RNA, the coarse-graining is achieved by investigating secondary structures instead of the full 3D structure of a molecule, proteins can be modeled by putting each monomer on a 2D or 3D grid and allowing only specific movements on this grid. Although such simplifications include a considerable level of abstraction, these models do not only give an impression of energetic properties of biomolecules. Simple models also allow for a thorough investigation of biopolymer folding properties as well as dynamics and kinetics studies that would not be possible within an exact model.

Several powerful algorithms that make a computational treatment of RNA feasible have been suggested within the last decades [117, 157, 170], a freely available implementation of these algorithms is the `Vienna RNA Package` [77, 78]. In contrast to RNA, efficient algorithms to calculate the ground state of protein models are not available. The situation is even worse here, especially due to the fact that lattice heteropolymer folding was shown to be NP-complete [9, 30, 151].

It is necessary to get an impression of the underlying energy landscape in order to investigate the dynamic behavior of a biomolecule. For RNA, efficient dynamic programming tools for the calculation of all suboptimally folded secondary structures within a desired energy range above the ground state are available [164]. For lattice protein models, the set of suboptimal structures must be enumerated exhaustively, preventing an examination of reasonably long model chains. Nevertheless, with knowledge about all suboptimal structures, an insight into the energy landscape of biopolymers is possible (for a thorough introduction to characterization and computation of general landscapes, see [49, 143]). Another prerequisite to investigate energy landscapes is some sort of metric that defines adjacency between different (secondary) structures, called *move set*. To be more precise, a move set is an order relation on the set of conformations that influences the shape of the energy landscape dramatically: Depending on which combinations of elementary moves are allowed, the energy landscape can be very rugged or more or less smooth. While RNA landscapes are thought to be extremely rugged, energy landscape of simple protein models are generally highly degenerate - this can basically be seen as an artefact of the assumed simplification in the model.

The energy landscape of a biopolymer is determined by (a) the set of configurations of the molecule, (b) the move set and (c) an energy function that assigns an energy value to each legal configuration. The properties and topology of the underlying energy landscape influence the folding behavior of a biomolecule. A number of techniques for calculation of features such as the number of local optima, the size distribution of the basins of attraction as well as a practical visualisation in form of so called *barrier trees* (see chapter 4.1) have been developed for the special case of RNA secondary structures within the last years [49]. Similar tools can be applied to lattice protein folding. We present here a novel approach to enumerate the near-ground state part of the energy landscape (see chapter 6).

Having the energy landscape (or at least its low-energy part) at hand, the dynamics of a biomolecule can be investigated by means of a reduced description of the state space. In contrast to a previously suggested Monte Carlo method that considers *every* single configuration of the interesting molecule [47], we present here a recently published method that assesses the dynamics by modeling it as a Markov process on the level of barrier trees (see chapter 7). Within this model, only local minima of the barrier tree are allowed states of the system. Transition rates between these minima depend on the energy barriers between them. Fractional population densities are assigned to certain local minima at the beginning of the simulation. Depending on the initial conditions and the energy ratio in terms of the barrier height between different states, other states (local minima) are populated as time elapses. This method allows not only for a thorough and fast study of the *whole* dynamics of a biopolymer, but enables also investigation of the refolding behavior of biomolecules on a theoretical level. More generally, properties of the folding landscape such as kinetic traps can easily be found with this model.

2 Biopolymer Modelling

Biopolymers are polymers found in nature. DNA, RNA, proteins and polysaccharides are examples of biopolymers in which the monomer units, respectively, are nucleotides, amino acids and carbohydrates. We will give an introduction to RNA and proteins in this chapter as they form the molecular basis of all calculations presented in this thesis.

2.1 RNA

Ribonucleic acid (RNA) is a linear polymer with a backbone of ribose sugar rings linked by phosphate groups. Each sugar has one of the four naturally occurring bases adenine (**A**), guanine (**G**), cytosine (**C**) and uracil (**U**) linked to it as a side group. The sequence of these bases specifies the structure and function of a RNA molecule. The 5' carbon of one ribose is linked to the 3' carbon of the next ribose via a phosphate group, hence the backbone is directed: Since one end has an unlinked 5' carbon and the other one an unlinked 3' carbon, the ends are referred to as 5' and 3' ends. The chemical difference between RNA and DNA is small: RNA has an OH-group at the 2'-position of the ribose ring, DNA has just a H bound there. Further, DNA contains thymine (**T**) instead of uracil (**U**). However, RNA and DNA differ in their structure: Whereas DNA mostly occurs in double-stranded, perfectly complementary helical structures, RNA usually occurs single-stranded. In other words we can say that when RNA is transcribed in cells as single strands of nucleic acids, these are not simply long strands of nucleotides. Rather, intra-strand base pairing produces a complex arrangement of structure motifs.

It has been thought for a long time that there is a strict partitioning in the way genetic information is processed: DNA is used for storage, RNA acts as a transmitter and proteins are regarded chemical catalysts. This so called 'central dogma of molecular biology' has considered RNA as merely an intermediate between DNA and proteins. However, RNA turned out to be 'more': RNA molecules do not only serve as carriers of information, but also as functionally active units, i.e. RNA is seen nowadays as an important and versatile molecule on its own. See [74] and references therein for an in-depth overview of RNA structure and function.

mRNA (messenger RNA) is an exact copy of one of the strands of a certain region of DNA and its central region serves for a template during protein synthesis. tRNAs (transfer RNA), figure 1, that have been sequenced in many organisms, are short sequences of about 76 nucleotides that form a well-defined clover-leaf structure. They play a crucial role in the process of protein assembly as they are charged with an amino acid at the 3' end which is incorporated into the nascending peptide chain during protein synthesis. We did some kinetic studies with $tRNA^{phe}$, see section 7.3.1. RNA is also present at another site of protein synthesis: Ribosomes are composed of two sub-units, each containing three types of rRNA (ribosomal RNA) as well as several different proteins. Ribosomes have binding sites for mRNA as well as tRNA and they move sequentially along a mRNA template, acting on one codon at once.

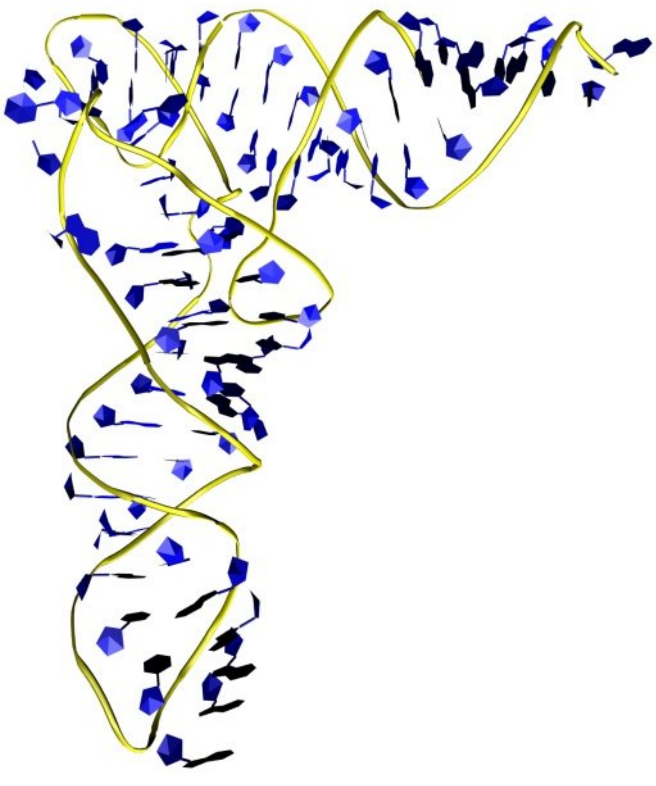


Figure 1: Tertiary structure of $tRNA^{phe}$ from yeast (*saccharomyces cerevisiae*). The backbone is drawn as yellow ribbon, purines an pyrimidines of the nucleotides are shown in blue. tRNA occurs in typical L-shaped structures, the anticodon loop at the bottom is the counterpart to the respective codon on an mRNA. The tRNA is loaded with an appropriate amino acid at the CCA motif (right top) by the enzyme aminoacyltransferase.

When Cech and Altman discovered that RNA has the ability to exhibit catalytic activity [23, 60, 61], the idea that an *RNA World* [54, 82, 83] stood at the origin of life was born. Within this view, RNA served both as carrier of genetic information as well as catalytically active substance. RNA may not necessarily have been the first step in prebiotic evolution, but the idea that RNA preceded not only DNA, but also the invention of the translational system, seems widely accepted. While the activity of these so called *ribozymes* is usually restricted to cleavage and splicing of RNA itself, recent evidence suggests that RNA also plays a predominant role in ribosomal translation. DNA molecules exhibiting catalytic behavior have also been discovered [20].

Another interesting example of RNA involvement are *RNA viruses*, particles consisting of one or more RNA molecules contained within a protein coat. Viral RNA genomes not only code for proteins, but also carry out the role normally adopted by DNA in storing genetic information. Many different families of RNA viruses are known: Simple bacteriophages, such as Q_β or MS2, which multiply inside bacterial cells. More complex examples include plant pathogens like Tobacco Mosaic Virus or human pathogens like influenza or HIV. RNA secondary structure motifs are known to play a crucial role in the viral cell cycle. Well known examples are the internal ribosomal entry site (IRES), the RRE motif in HIV or the CRE hairpin in Picornaviridae. A comprehensive survey of structural features across the full genomes of the whole family Flaviviridae was given recently [147].

Newer investigations revealed that *riboswitches* - complex folded domains that serve as receptors for specific metabolites - play a crucial role in controlling genes [46, 48, 99, 105, 116, 124, 160]. Generally, gene-control systems must have the ability to respond precisely to specific signals, rapidly bring about their genetic effect and have sufficient dynamic character to fine-tune the level of expression for hundreds of different genes. Although it has long been realized that protein-based control systems are present in organisms, there is an emerging awareness of the role of RNA factors in gene control nowadays: On the one hand side RNA is present in gene control mechanisms in form of microRNAs (miRNAs) and related short-interfering RNAs (siRNAs). These are short non-coding RNA fragments of about 22 nucleotides in length that regulate gene expression by several mechanisms like post-transcriptional gene-silencing or DNA methylation [43, 68]. Although protein factors are required for proper operation of these mechanisms, many organisms rely on RNA for critical regulatory tasks.

On the other side numerous mRNAs in procaryotes exhibit complex folded domains within the non-coding region of their nucleotide chain that directly bind to specific metabolites. These *riboswitches* control gene expression by harnessing changes in RNA structure without involvement of protein factors, influencing transcription elongation, translation initiation or other aspects of the process that leads to protein production [116].

It has been shown repeatedly that alternative conformations of the same RNA sequence can perform completely different functions, see e.g. [6, 120, 135]. SV11, for instance, is a relatively small molecule that is replicated by Q_β replicase. It exists in two major conformations, a meta-stable multi-component structure and a rod-like conformation, constituting the native state, separated by a huge energy barrier. While the meta-stable conformation is a template for Q_β replicase, the ground state is not. By melting and rapid quenching the molecule can be reconverted from the inactive stable to the active meta-stable form [166].

In recent years dynamical aspects of RNA structure formation, including transitions at the level of RNA secondary structure, have received increasing attention, because they can play a crucial role for the understanding of the biological function of RNA. It has been shown for a number of natural RNAs that the formation of alternative or metastable conformations are well-defined steps in their folding pathways. These folding intermediates determine the biological function of the molecule.

The translation of the four genes encoded on the genomic RNA of the bacteriophage MS2 is regulated by the secondary structure transition of the 5' untranslated leader sequence from a metastable hairpin to a stable cloverleaf structure [122]. While the expression of the lysis and replicase genes is coupled to the expression of the coat protein in the full-length RNA, the maturation gene, coding for the A-protein needed by the virion for the attachment to *E. coli*, is unaccessible to the ribosome due to the cloverleaf structure of the leader sequence. During transcription of the viral RNA the 5'-end of the leader sequence is trapped in a metastable hairpin allowing the ribosome to access the A-protein gene. After some time the hairpin is disrupted in favor of the stable cloverleaf, thereby silencing the A-protein gene expression. This secondary structure switch precisely controls the amount of A-protein translated from the MS2 genomic RNA.

The Hok/Sok system of plasmid R1 from *E. coli* is another prominent example for the regulation of gene expression via an intricate cascade of secondary structural rearrangements. The Hok/Sok system mediates plasmid maintenance by expressing the Hok toxin which kills plasmid-free segregates. The plasmid encodes for a highly stable mRNA, which is translated to the Hok toxin if the mRNA is in its activated conformation, and a labile anti-sense RNA (Sok) which act as an antidote by binding to the activated *hok* mRNA, leading to rapid degradation of the resulting duplex. The full-length *hok* mRNA forms a pool of inactive mRNAs. In time, however, the *hok* mRNA gets processed resulting in the truncation of the 3'-end, which triggers a refolding of the mRNA into the active conformation. Then both locations, the Hok gene and the Sok binding site are accessible. If the plasmid was lost, the pool of the antidote Sok is depleted, since the *hok* mRNA is considerably more stable then the *sok* RNA inducing the killing of the cell. For recent reviews on biologically functional RNA switches we refer to [12, 115, 105].

The structure formation process of RNA can conceptually be partitioned into two consecutive stages [14, 148]. First, the specific sequence (the string of bases) or *primary structure*, is transformed into a pattern of complementary base pairings called the *secondary structure*. Second the secondary structure distorts, to form a three dimensional spatial structure or *tertiary structure* (see section 2.4 for a formal definition of RNA secondary structure). The tertiary structure is the three-dimensional configuration of the molecule. Tertiary interactions are hydrogen bonding or stacking interactions between structure elements. Although the hierachical nature of RNA formation is generally accepted, there exist examples where the secondary structure is changed after the tertiary strucrure has been fomed [163].

It is hard to solve the structure prediction problem for RNA structures since the number of degrees of freedom of the RNA chain is very high. Nevertheless, there are several facts that support the consideration of the secondary structure of RNA as a coarse grained approach to the three dimensional spatial structure:

- The conventional base pairing and the base stacking cover the major part of the free energy of folding.
- The secondary structure provides a scaffold of distance constraints to guide the formation of the tertiary structure.
- In contrast to the protein case, the secondary structure of RNA is well

defined and assigns all bases to secondary structure elements.

- RNA secondary structure is conserved in evolution and has been used successfully to interpret RNA function and reactivity.

The secondary structure of RNA is formed by aggregation of planar complexes, or *base pairs* of purine and pyrimidine bases. There are four naturally occurring bases: Adenine (**A**), Guanine (**G**), Cytosine (**C**) and Uracil (**U**). **G** and **C**, respectively **A** and **U** are complementary bases which can form strong hydrogen bonds, a weaker base pair is also possible between **G** and **U**, often referred to as "wobble" base pair.

2.2 Proteins

Proteins are macromolecules that are constructed from one ore more unbranched chains of 20 different amino acids linked by peptide bonds. Amino acids can be hydrophobic or hydrophilic, small or large, charged or uncharged. A typical protein contains a few hundred amino acids, though short chains (the smallest are often called peptides) and extremely long ones are known. One of the largest proteins known at this time is *titin*, a protein found in skeletal and cardiac muscles, which contains on average 26926 amino acids in a single chain.

Proteins exhibit an extraordinary diversity of function. One protein is responsible for transport of oxygen in the blood, another produces a strong, fibrous structure found in hair and yet another one catalyzes cleavage of nucleic acids. The following list is an incomplete enumeration of protein function:

- nearly all biochemical reactions are catalyzed by enzymes that mostly contain proteins
- any form of motion in living cells is based on contractile proteins, e.g. muscle fibers
- the structure of cells, and the extracellular matrix in which they are embedded, is largely made of protein[1] (for example collagens)

[1] plants and microbes mostly rely on polysaccharides (cellulose) for support, but these are synthesized by proteins

- proteins are often found in signal-transduction mechanisms
- heterotrophic nutrition is crucially dependent on proteins
- proteins occur as transcription factors that turn genes on and off

In proteins, amino acids are connected via peptide bonds, where the carboxyl group of one amino acid is connected to the amino group of the other (figure 2). The sequence of amino acids is often called *primary structure*. In the 1930ies, Pauling an Corey found that the peptide bond C-N is somewhat shorter than the C-N in a simple amine and that the atoms associated with the peptide bond are coplanar. They found a resonance or partial sharing of two pairs of electrons between the carboxyl oxygen and the amide nitrogen. A small electric dipole is set up by a partial negative charge of the oxygen and a partial positive charge of the nitrogen.

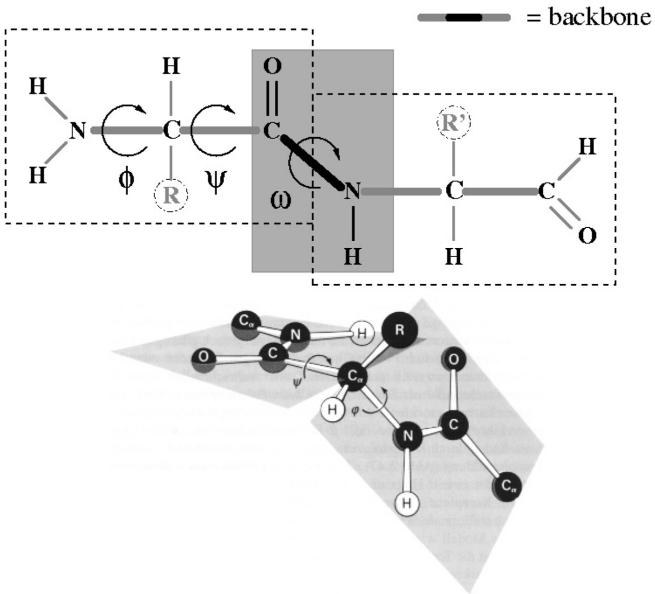


Figure 2: **Upper plot:** The peptide bond. Torsional angles are labeled with Greek letters. Rotation around ω is generally hindered. **Lower plot:** Three bonds separate sequential α carbons. Six atoms of a peptide group lie in single plane with the oxygen of the carboxyl group and the hydrogen of the amide nitrogen trans to each other. Rotation is only possible around the ϕ and ψ angles.

The lower plot of figure 2 shows that the backbone of a polypeptide chain can be pictured as a series of rigid planes with consecutive planes sharing a common point of rotation at C_α. Free rotation around the peptide bond is not possible, more flexibility for rotation is around the N-C_α bond (called ϕ angle) and around the C_α-C bond (called ψ angle). In principle, both ϕ and ψ can have values between -180° and +180°. Nevertheless many values are prohibited by steric interference between atoms in the polypeptide backbone and amino acid side chains. In real proteins, the achievable values are restricted to small regions that are displayed in so called Ramachandran plots.

The term *protein secondary structure* refers to a local conformation of parts of a polypeptide, i. e. a periodic spatial arrangement of residues that are close to each other on the (primary) amino acid chain. A few types of secondary structure are particularly stable and are found in may proteins, the most prominent are α helix and β conformations described below, though there are other secondary structure elements such as loops and turns that allow a polypeptide chain to change direction.

An *α helix* is a rod-like coiled structure where the polypeptide backbone is tightly wound around an imaginary axis drawn longitudinally through the middle of the helix. The inner part of the helix is formed by the backbone, while the side-chains are turned outward. In this arrangement, all non-terminal CO and NH groups are hydrogen bonded, which means that an α helix makes optimal use of internal hydrogen bonds.

A β conformation is a more extended conformation of the polypeptide where the backbone is structured in a zigzag layer rather than a helical structure. These layers (β strands) can be arranged side by side (parallel or anti-parallel) to form a structural element called *β sheet*. In this arrangement, hydrogen bonds are formed between adjacent segments of the polypeptide chain. Although we claimed protein secondary structure to refer to local conformations, the individual segments that form a β sheet can be quite distant from each other, they may even be segments in different polypeptide chains.

Protein *tertiary structure* will be of special importance for our purpose in this thesis. With this term we refer to the three-dimensional arrangement of all atoms in a protein. In contrast to secondary structure (referring to local arrangements of amino acids), tertiary structure includes longer-range aspects of amino acid sequence. Amino acids that are spatially far apart from each other and that reside in different secondary structures can interact within the completely folded structure. Weak-bonding interactions or covalent bonds such as disulfide cross-links are often responsible for the tertiary positions of interacting segments in the polypeptide chain.

2.3 Continuous Space models - State of The Art

We gave an introduction to biopolymers, their function and current understanding in the last sections. The main intention of this thesis is investigation of the *dynamics* of biopolymers. Despite all the progress in recent years [121, 138], it seems fair to say that Molecular Mechanics will for the foreseeable future remain incapable of predicting, say, the folding pathway of a globular protein starting from a random coil state all the way to its (unknown) native state. Thus, we will present a different approach here: Reduced models that are simple representations of biopolymers. In the following we will give a brief introduction on the concepts of and prerequisites necessary to modeling biopolymers within a computational framework.

In order to treat this class of molecules on a theoretical level and investigate its behavior computationally we need some sort of abstraction. The most evident difference between modeling nucleic acid and protein dynamics is the *level* of abstraction used as a basis for the models: In contrast to protein folding, the secondary structures of nucleic acids provides a level of description that is sufficient to understand the thermodynamics and kinetics of RNA folding [146] (without recourse to an atom-by-atom model of the molecule), see section 2.4. Before giving an introduction to the "simple-exact" methodology of protein folding (section 2.5), we start with an overview of generally known computational continuous-space methods that are readily applicable to proteins and nucleic acids.

Given, that computational resources would not represent the limiting factor, quantum mechanical methods would be a first choice for studying conformations and interactions of biomolecules. These methods solve for the electronic structure of molecules and thus derive the effective Born-Oppenheimer potential for nuclear motion from first principles. However, such methods are enormously resource- and time consuming for larger biomolecules. Instead, force field methods that ignore the electronic motions are used to calculate the energy of systems as a function of the nuclear positions only.

The principles of force fields (also known as molecular mechanics) are based upon Newtonian mechanics. The basic idea is that bond lengths, valence and torsional angles have "natural" values depending on the involved atoms and that molecules try to adjust their geometries to adopt these values as closely as possible. Additionally, steric and electrostatic interactions, mainly represented by van der Waals and Coulomb forces, are included in the so-called potential. A typical force field contains a set of several potential functions which themselves contain adjustable parameters. These parameters are optimized to obtain the best fit of experimental values, as geometries, conformational energies and spectroscopic properties. It is important to realize that force fields are usually parameterized for a limited set of molecular properties and a specific set of molecules.

Many of the molecular modeling force fields in use today can be interpreted in terms of a relatively simple four component picture of intra- and intermolecular forces within the system (see Appendix B for details).

$$E_{total} = E_{bond} + E_{angle} + E_{torsion} + E_{non-bonding}$$

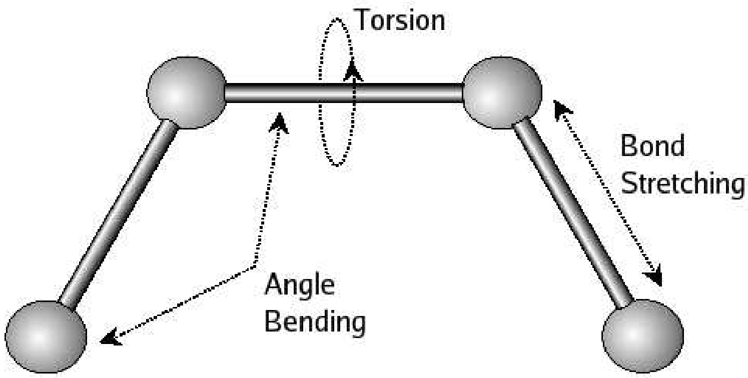


Figure 3: Three components picture of molecular forces. Non-bonding forces are not shown.

These simple terms mentioned above can be expanded to adjust the potentials better to the experimental results (e.g. Morse potential for bonds, Taylor expansions with higher terms, cross-terms between the potentials), but with the disadvantage of higher calculational effort. That is the reason why biomolecular force fields usually do not include refinement terms for the bond, angle and torsion potential. Sometimes force fields include additional potential terms for specific interactions, such as hydrogen bonding or dipole-dipole interaction. A typical example is the hydrogen bonding term in the AMBER force field.

Calculating the energy with respect to a given conformation is only one part of optimizing the structure of molecules. For improving the structure it is necessary to change the geometry in such a way, that the total energy is lowered. This process is repeated iteratively so that an energy minimization corresponds to a geometry optimization. Ideally, finding the **global minimum** of the underlying potential function is desired. Unfortunately, there is no method available to determine the global minimum of a function of many variables. Hence, optimization algorithms (steepest descent, conjugate gradient) are often trapped in a **local minimum**. A consequence of ending the optimization run in a local minimum is that different optimized structures will be achieved, depending on the starting geometry. Therefore it is usually necessary to use different starting geometries and compare the obtained structures to get lower energies.

A useful approach to overcome the problem of local trapping is the implementation of some kind of randomness (namely stochastic techniques) as it is done with **simulated annealing**. Simulated annealing is a widely used optimization procedure that originated in statistical physics [85]. In effect it tries to simulate the cooling and the crystallization process occurring in a heated solid. Starting point is the configuration space Ψ and an energy function U, which $U : \Psi \rightarrow \mathbb{R}$. In the case of molecular mechanics U corresponds to the potential function whereas Ψ is the conformation space constructed from all possible conformations of the molecule. Beginning from a starting geometry, the energy E_0 of the molecule is calculated. This is followed by a random step in conformation space, representing a random change of the molecular geometry. Then the energy E_1 of the new conformation is calculated. The probability p of accepting the new conformation as a new starting structure at this point is given by:

$$p = \begin{cases} 1 & : \quad E_1 \leq E_0 \\ e^{-\frac{E_1 - E_0}{kT}} & : \quad E_1 > E_0 \end{cases}$$

k and T are the Boltzmann constant and the temperature, respectively. In fact, this is known as the Metropolis algorithm [110]. It ensures that the optimization cannot be trapped in a local minimum since higher energies are accepted with a certain probability so that energetic barriers can be overcome. If n is the number of simulated annealing steps the global minimum is always found for $n \rightarrow \infty$. A typical simulated annealing procedure starts at high temperature T to warrant that the random walk overcomes the highest barriers and reaches most of the conformational space. Then the temperature is lowered by a certain scheme and the molecule is trapped in the conformation it has entered most often.

In **molecular dynamics** (MD), successive configurations of a system are generated by integrating Newton's laws of motion. The result is a trajectory that specifies how the positions and velocities of the particle in the system vary with time.

$$F_i(t) = m_i a_i(t) = m_i \frac{\partial^2 r_i(t)}{\partial t^2}, \text{ where } F_i(t) = -\frac{\partial E_{tot}}{\partial r_i}$$

The forces acting on the atoms are the negative gradient of the potential energy E_{tot}. Under the influence of a continuous potential the motions of all the particles are coupled together, giving rise to a many-body problem that cannot be solved analytically. Therefore the equations of motion are integrated using a *finite difference method*. As basic idea the integration is broken down into small stages, each separated in time by a fixed time δt. The accelerations a_i of the particles are available from the force F_i, calculated from E_{tot}. The accelerations a_i are then combined with the positions and velocities at a time t to calculate the positions and velocities at a time $t + \delta t$. Choosing an appropriate time step δt is essential for a successful molecular dynamics simulation. Typical δt for all-atom force fields with no constraints is 1 femtosecond. As the process of folding takes place in a millisecond scale, the simulation of biomolecular folding by atomistic MD is not within the reach of present day computers.

There are several force field program packages available for biomolecular computation. The most prominent of these force fields is the Cornell force field of AMBER, which is not only used in the AMBER packages, but is also included in various other program packages (e.g. NAB, JUMNA). Other examples of force fields are CHARMM (**C**hemistry at **HAR**vard **M**olecular **M**echanics) [15, 101] and GROMOS (**GRO**ningen **MOL**ecular **S**imulation System) [154]. The potentials in AMBER, CHARMM and GROMOS have the same basic structure as described in the beginning of this section. Only AMBER has an additional energy term for an adequate description of hydrogen bonds. Apart from these force fields there are some other packages for simulating biomolecules including other energy term expressions, like DREIDING [107] and Tripos 5.2 [28].

2.4 RNA Secondary Structure

We consider nucleic acid structures at a coarse-grained level, representing each nucleotide by a single point. Only covalent and non-covalent contacts (the latter correspond to specific hydrogen bonds) are used instead of spatial coordinates, hence only RNA sequence and the list of base pairs enter our considerations.

A secondary structure S is formally defined as the set of all base pairs (i, j) with $i < j$ such that for any two base pairs (i, j) and (k, l) with $i \leq k$ the two following conditions hold [157]:

1. $i = k$ if and only if $j = l$.
2. There are no knots or pseudoknots allowed. For any two base pairs (i, j) and (k, l) the condition $i < k < l < j$ or $k < i < j < l$ must be satisfied.

The first condition simply means that each nucleotide can take part in at most one base pair. Several examples of tertiary interactions breaking this condition are known, including base triplets, G-quartets and A-platforms. The second condition forbids knots and pseudoknots. While pseudoknots are important in many natural RNAs [159], they can be considered part of the tertiary structure for our purposes and we will therefore neglect them for the purpose of this thesis.

The two conditions above imply that secondary structures form a special type of graphs. In particular, a secondary structure graph is *sub-cubic* (i.e. the vertex degree is at most three) and *outer-planar*. The latter property means that the structure can be drawn in the plane in such a way that all vertices (representing nucleotides) are arranged on a circle (the molecule's backbone), and all edges (representing base pairs) lie inside the circle and do not intersect, see figure 4.

The possibility to compute the free energy of structure formation given the sequence and the list of base pairs forms the physico-chemical basis for a coarse grained computational chemistry of nucleic acids. Note that a secondary structure as defined here corresponds to an *ensemble* of conformations restricted to a certain base pairing pattern. No information is assumed about the spatial conformation of unpaired regions. Since the entropic contributions of these restricted conformations have to be taken into account, we are dealing with (temperature dependent) *free energies* here.

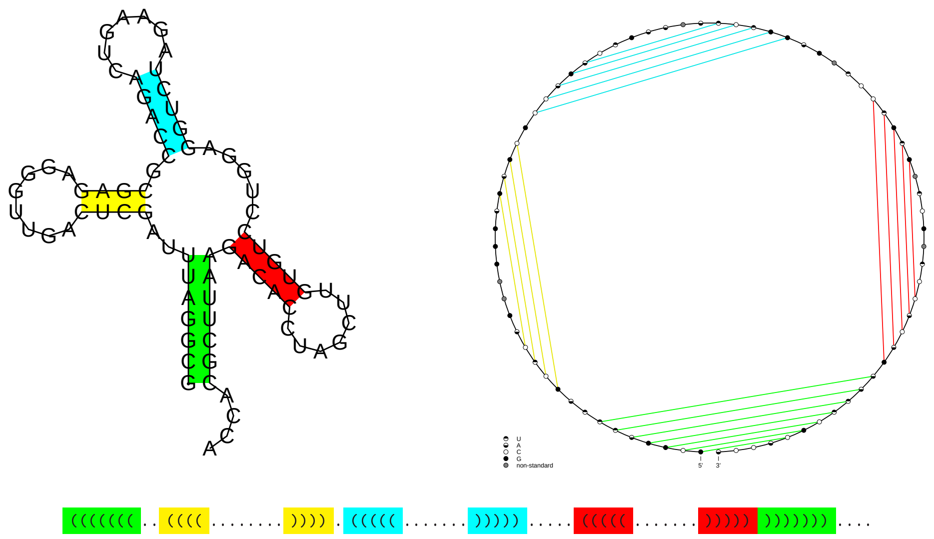


Figure 4: Secondary structure of phenylalanine-tRNA from yeast as conventional drawing and in circular representation. The chords in the circular representation must not cross in secondary structure graphs.

Any secondary structures can be uniquely decomposed into loops as shown in figure 5 (note that a stacked base pair may be considered as a loop of size zero). A secondary structure graph is equivalent to an ordered rooted tree. An internal node (black) of the tree corresponds to a base pair (two nucleotides), a leaf node (white) corresponds to an unpaired nucleotide. Contiguous base pair stacks translate into “ropes” of internal nodes, and loops appear as bushes of leaves.

Both quantum chemical calculations and thermodynamic measurements suggest that horizontal (base pairing) contributions to the total energy depend on the base pair composition, whereas vertical (base stacking) contributions depend on base pair composition *and* base sequence, i.e. the upstream and downstream neighbors along the chain [132]. The *nearest neighbor model* introduces the assumption that the stability of a base pair, or any other structural element of a RNA, is dependent only on the identity of the adjacent base and/or base pairs. The model is justified by the major contribution of short-range interactions (hydrogen bonding, base stacking) to the overall stabilizing energy of nucleic acid structures. In addition, it is natural to assign loop entropies instead of individual bases. Stacks are treated as special types of loops. The energy of an RNA secondary structure S is thus assumed to be the sum of the energy contributions

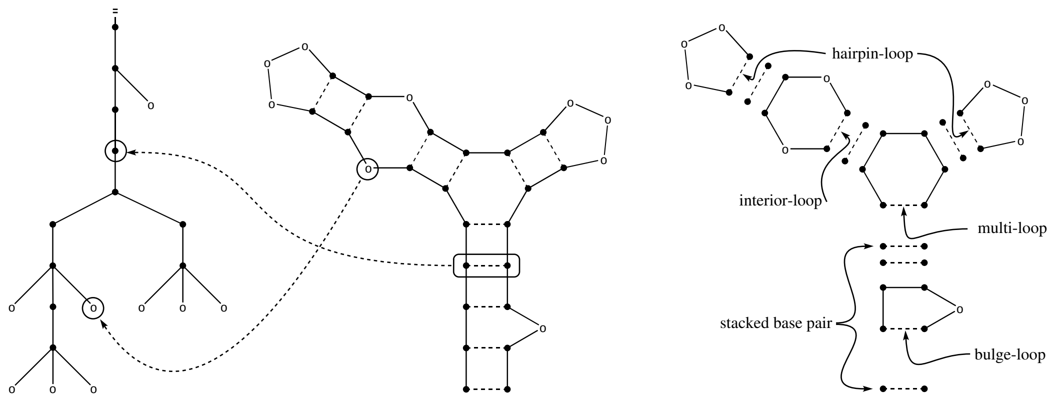


Figure 5: Various representations of RNA secondary structure: The tree representation of the secondary structure graph in the middle (l.h.s); Representation of an RNA secondary structure as a planar graph (middle); The loop decomposition of the secondary structure graph in the middle (r.h.s). The closing base pairs of the various loops (base pair, hairpin, bulge, interior, multiloop) are indicated by dotted lines (Note that a helix of length n decomposes in n-1 stacked base pairs).

of all "loops"[2] L

$$E(S) = \sum_{L \in S} \epsilon(L) + \epsilon(L_{ext}) \tag{1}$$

where L_{ext} is the contribution of the "exterior" loop containing the free ends. Note that stacked pairs are treated as minimal loops here. This decomposition has a solid graph theoretical foundation [96]: the loops form the unique minimal cycle basis of the secondary structure graph. More importantly, a large number of careful melting experiments have shown that the energy of structure formation (relative to the random coil state) is indeed additive to a good approximation, see e.g. [51, 81, 106, 156].

Usually, only Watson-Crick (**AU**, **UA**, **CG** and **GC**) and wobble pairs (**GU**, **UG**) are allowed in computational approaches since non-standard base-pairs have in general context-dependent energy contributions that do not fit into the "nearest-neighbor model". Qualitatively there are two major energy contributions: Stacking of base pairs and loop entropies. Stacking energies can be computed for molecules in the vacuum by means of standard quantum chemistry approaches, see e.g. [75, 119]. The secondary structure model, however, considers only energy differences between folded and unfolded states in an aqueous

[2] i.e. the faces of the planar drawing of the structure

solution with rather high salt concentrations. As a consequence one has to rely on empirical energy parameters. Loops are destabilizing: The closing base pair restricts the possible conformations of the sequence in the loop relative to the conformations that could be formed by the same sequence segments in a random coil resulting in an entropy loss and thus an increase in free energy.

Let $\mathcal{A}$ be some finite alphabet of size κ, let Π be a symmetric Boolean $\kappa\times\kappa$-matrix and let $\Sigma = [\sigma_1 \ldots \sigma_n]$ be a string of length n over $\mathcal{A}$. A secondary structure is *compatible* with the sequence Σ if $\Pi_{\sigma_p,\sigma_q} = 1$ for all base pairs (s_p, s_q). Following [80, 157] the number of secondary structures $\mathcal{S}$ compatible with a specific string can be enumerated as follows: Denote by $S_{p,q}$ the number of structures compatible with the substring $[\sigma_p \ldots \sigma_q]$. Then

$$S_{l,n+1} = S_{l,n} + \sum_{k=l}^{n-m} S_{l,k-1} S_{k+1,n} \Pi_{\sigma_k,\sigma_{n+1}} \tag{2}$$

A secondary structure compatible with a given sequence with maximal number of base pairs can be determined by a dynamic programming algorithm [117]. Other variants of the algorithm have been formulated besides this "maximum matching" problem: Zuker and Stiegler [169, 170] formulated the algorithm for the minimum energy problem using the now standard energy model. Since then several variations have been developed: Zuker further devised a modified algorithm that can generate a subset of suboptimal structures within a prescribed increment of the minimum energy [168]. The algorithm will find any structure S that is optimal in the sense that for every pair b in S there is no structure S_b that contains the pair b and has lower energy than S.

John McCaskill [108] showed that the partition function over all secondary structures

$$Z = \sum_S \exp(-\Delta G(S)/kT) \tag{3}$$

can be calculated by dynamic programming as well. In addition his algorithm can be used to calculate the frequency with which each base pair occurs in the Boltzmann weighted ensemble of all possible structures. Having the partition function at hand, it is possible to explore the thermodynamics of RNA secondary structures. The free energy of structure formation, for example is, $\Delta G = -RT \ln Z$. From this, other thermodynamic parameters, such as melting curves, can be computed.

A for academic use freely available implementation of the just mentioned algorithms is the `Vienna RNA Package`[3] [76, 77, 78] which forms the basis of all RNA-related calculations presented here. Energy parameters used in the software can be found in [106].

Pseudoknots must be treated separately, since they do not meet the second condition given at the beginning of this section. The prediction of RNA pseudoknots, however, is still largely an open problem. Themodynamic structure prediction based on the standard energy model is NP-complete [1, 100] in general, albeit restricted classes of pseudoknots can be dealt with by polynomial algorithms. Nevertheless, these approaches are expensive in terms of CPU and memory usage [1, 38, 71, 126, 130, 131] and in addition suffer from uncertainties of the energy model for pseudoknots [62].

The *conformation space* X of a given sequence is the set of all secondary structures S compatible with this sequence. As mentioned earlier, each secondary structure $S \in X$ itself is a list of base pairs (i, j) in a way, that any two base pairs from S do not cross each other, if S is represented as a graph in the plain. From the total recursion (equation 2) an asymptotic formula for the growth of the number of secondary structures with chain length n can be derived [19, 40, 79, 109].

$$S_n \sim n^{-\frac{3}{2}} \cdot \alpha^n \tag{4}$$

Counting only those planar secondary structures that contain hairpin loops of size three or more (steric constraint), and that contain no isolated base pairs one finds $\alpha = 1.8488$ for the total number of secondary structures. The size of the conformation space increases exponentially with the chain length. A convenient measure to get am impression on the conformation space X of a given sequence is the density of states $g(\varepsilon)$. It displays the energies of the individual structures S, and their distribution with regard to the ground state. Algorithms for computing the complete density of states for a given RNA sequence are available [32]. However, the density of states gives only the number of conformations in a certain energy range, but not their explicit structures. If we were interested in this information, suboptimal folding techniques are needed. We used the tool `RNAsubopt`[4] [164] which, based on an efficient dynamic programming algorithm, provides *all* suboptimal folds for a given sequence within a desired energy range.

[3]available from `http://www.tbi.univie.ac.at/~ivo/RNA/`

[4]part of the `Vienna RNA package`

2.5 Discrete Protein Models

As mentioned before, structure calculations of proteins at an atomic resolution make use of the same force field machinery as nucleic acdis' calculations do. Within continuous space models, a crucial problem is of course the large number of degrees of freedom. Consequently, it is still impossible to determine the minimum energy structure for larger proteins based on the knowledge of only their sequence. To circumvent this problem, many approaches were made to reduce the conformation space, Most of them work with reduced amino acid representations on various lattices. The simplest approaches use only one representative pseudo atom per amino acid (mostly Cα sometimes Cβ), extended versions include additional pseudo atoms for the side chains. Lattice models have several advantages. First, such models can explore larger conformational changes and they allow for an easy design of local conformational transitions. In contrast to conventional MD simulations, lattice models enable precalculation of entire sets of some conformational transitions. Atomic-level simulations can currently explore only small conformational changes that occur within very short times. Second, lattice models overcome the problem of incomplete sampling that is inherent in atomic resolution models (due to parameters and approximations that must be assumed there). Third, lattice models do not include terms for covalent energies and thus circumvent the problem of calculating small differences (few kilocalories) between large energy terms (megacalories) [35].

There are in principle two types of lattice model simulations, aiming at two distinct objectives. One was designed to understand the basic physics governing the protein folding process. The key feature of this lattice type is its simplicity. The energy evaluation on such a lattice model can be achieved quite efficiently. Based on this type of models, methods involving exhaustive searches of the available conformation space became feasible. However, most of these models are unable to describe subtle geometric aspects of proteins' conformation. Prominent examples of this type are the models of Gō and coworkers [55] and the HP-model proposed by Dill [33, 35]. Gō models were used to study folding kinetics using hypothetical potential functions (intrachain attractions are only considered if a pair of monomers is arranged in its native conformation) with Metropolis Monte Carlo sampling in simple lattice models. Gō models suffer from sparse sampling and the unphysical potential. Dill and collaborators provided a framework that accounts for more realistic features of heteropolymers: Amino acids are divided into two categories (hydrophobic **H** and polar **P**). The polypeptide chain was originally modeled on square (`SQ`) and simple cubic (`SC`) lattices, respectively. Although the simplicity makes it possible to study the model in great detail, the main weakness of this model is related to the lack of a clear notion of secondary structure. Attempts were made to enable a better description of local structure by modeling the polypeptide chain by means of a symbolically defined secondary structure on tetrahedral (`TET`) [167], body-centered-cubic (`BCC`) and face-centered-cubic (`FCC`) lattices. A common feature of all studies addressing the HP model is that they claim to being *simple, yet exact*. This should illustrate that these models account for a hydrophobic collapse, enable some form of folding kinetics and design of foldable sequences with unique ground-states and enable modeling a two-state cooperativity in the folding process. Much effort was put into studying the HP model by different means within the last 15 years. (We just refer to some relevant contributions of Šali et al. [153], Shakhnovich et al. [111] and Karplus et. al. [37, 152] here.) See sections 2.6, 2.7 and chapter 3 for our implementation of this model.

Lattice models by Skolnick et al. [139], Miyazawa and Jernigan [113, 114] belong to the second category of lattice models. These models are geared towards realistic folding of real proteins. They are parametrized using measured protein structures. By statistical sampling of such available structures model templates are created. The resulting potentials are often referred to as statistical potentials. Works by Crippen [31], Eisenberg at al. [11] and Sippl et al. [73] are further examples of this category.

Both approaches can be uncoupled from the lattice condition, resulting in the so called *off-lattice models*. The origin of off-lattice models can be found in the works of Warshel and Levitt [94, 95]. In the simplest approaches the protein is represented by a chain of balls (amino acids) connected via stiff bonds. All energy functions used in lattice models have also been used in off-lattice models (e.g. [118, 137]).

For a recent review of reduced protein models we refer to [88]. Two review articles by Dill address the whole framework of heteropolymer modeling along with protein folding theory [34, 35].

2.6 Lattices and Self-Avoiding Walks

Let $\{\nu_1, \ldots, \nu_d\}$ be a set of d linear independent vectors in $\mathbb{R}^m$, $d \leq m$. A *lattice* is a set

$$\mathcal{L} = \{\vec{x} \in \mathbb{R}^m | \vec{x} = \mathbf{M}\xi, \xi \in X \subseteq \mathbb{Z}^d\}$$

where the matrix $\mathbf{M}$ with columns ν_1 through ν_m is called *generating matrix* of the lattice. The dimension of the lattice is $\dim\mathcal{L} = d$. It is always possible to represent a lattice by a square generating matrix. As an example consider the *face centered cubic* lattice `FCC` (figure 6). A generating matrix for `FCC` is

$$\mathbf{M}_{\texttt{FCC}} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix}$$

The corresponding set of integer vectors X is the set of all triples with an even sum. It is always possible to find a representation such that $\vec{0} \in \mathcal{L}$. We will denote $\vec{0}$ the origin of the lattice. Once $\mathbf{M}$ is defined it is computationally advantageous to perform all calculations in terms of the integer vectors $\xi \in X$ instead of the lattice points.

An *automorphism* of a lattice $\mathcal{L}$ is a distance-preserving transformation (isometry) of the space $\mathbb{R}^n$ that fixes the origin $\vec{0}$ and maps the lattice onto itself. The automorphisms of $\mathcal{L}$ form the group $\mathbf{Aut}[\mathcal{L}]$. It is sometimes useful to consider the group of *all* distance preserving maps that map $\mathcal{L}$ to itself, the so-called *affine automorphism*. This group is obtained by adjoining the translations in lattice vectors to $\mathbf{Aut}[\mathcal{L}]$.

We want to model linear polymers on a given lattice $\mathcal{L}$ by placing monomers at adjacent lattice points. This adjacency and the resulting neighborhood relation remain to be specified. The simplest way is to use nearest neighbors with respect to euclidean distance in $\mathbb{R}^m$, although other choices yield interesting models as well: the knight-move lattice `KM` (figure 6) and its 3D derivative `TDKM` are defined via moves that do not lead to nearest neighbors in euclidean space [56]. We define the set of lattice points $\mathcal{N}(\vec{x})$ that are accessible by a single step from $\vec{x}$ and claim symmetry of this neighborhood relation: $\vec{y} \in \mathcal{N}(\vec{x}) \Longrightarrow \vec{x} \in \mathcal{N}(\vec{y})$. As a consequence we can regard the lattice as an undirected graph $\Lambda = (\mathcal{L}, \mathcal{N})$, the vertices being the lattice points $\mathcal{L}$, the edges being defined by $\mathcal{N}$[5]

[5]Note that although the lattice points of the knight-move lattice are the same as those of

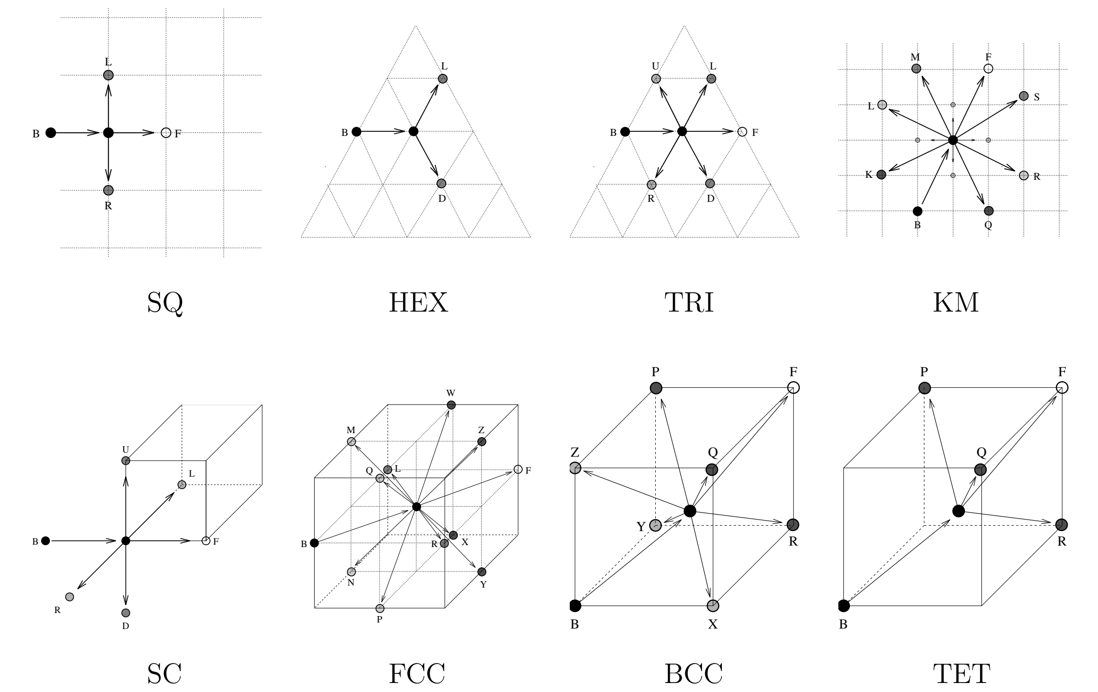


Figure 6: Lattices and relative moves. From top left to bottom right: Square `SQ`, hexagonal `HEX`, triangular `TRI`, knight's move `KM`, simple cubic `SC`, face-centered cubic `FCC`, body-centered cubic `BCC` and diamond `TET`.

Since we consider only transitive lattices we shall furthermore insist that the neighborhood $\mathcal{N}$ is the same everywhere on the lattice: We require that for any $\vec{x} \in \mathcal{L}$ there is an affine automorphism $\hat{\alpha}$ such that $\hat{\alpha}(\vec{x}) = \vec{0}$ and $\hat{\alpha}(\mathcal{N}(\vec{x})) = \mathcal{N}(\vec{0})$. As an immediate consequence, $\Lambda = (\mathcal{L}, \mathcal{N})$ is vertex transitive and thus regular. Each lattice point has exactly $z \stackrel{\text{def}}{=} |\mathcal{N}(\vec{0})|$ neighbors. We will further assume that the graph Λ is connected. In the following sections we will use the term lattice instead of the more exact "lattice graph" for the point set $\mathcal{L}$ embedded in $\mathbb{R}^m$ together with the adjacency relation $\mathcal{N}$. Figure 6 shows a subset of lattices we considered throughout our work.

A *walk* of length N on a lattice is a sequence $w = (\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots \vec{x}_N)$ of lattice points $\vec{x}_i \in \mathcal{L}$ such that $\vec{x}_i \in \mathcal{N}(\vec{x}_{i-1})$ for all $1 \leq i \leq N$. The number of distinct lattice points visited along a walk w will be denoted by $|w|$.

A walk w is *self-avoiding* if $\vec{x}_i \neq \vec{x}_j$ for all $i \neq j$. In other words, w is self-avoiding if and only if $|w| = N + 1$. Self-avoiding walks have a long history as

the square lattice, $\mathcal{L}_{\mathtt{KM}} = \mathcal{L}_{\mathtt{SQ}}$, their neighborhood relations are different, however.

models of linear polymers, as they incorporate the most important property of such molecules: excluded volume [39, 50]. For linear polymers $|w|$ equals the number of n of monomers, i.e., the *chain length*.

In the reminder of this section we show that a walk on a given transitive lattice graph $\Lambda = (\mathcal{L}, \mathcal{N})$ can be uniquely described by strings. We insist that w starts at the origin $\vec{0}$. We begin by choosing an arbitrary lattice point $\vec{\beta} \in \mathcal{N}(\vec{0})$ which we shall identify as the "backwards direction" of a walk. Next we assign an affine automorphism $\hat{\alpha}$ to each $\vec{\alpha} \in \mathcal{N}(\vec{0})$ such that (i) $\hat{\alpha}(\vec{\alpha}) = \vec{0}$ and (ii) $\hat{\alpha}(\vec{0}) = \vec{\beta} \in \mathcal{N}$. In general the choice of the automorphism $\hat{\alpha}$ is not unique; for our purposes this ambiguity does not have consequences. Of course we can write

$$\hat{\alpha}(\vec{x}) = \mathbf{S} \cdot (\vec{x} - \vec{\alpha}) \tag{5}$$

and thus it may be regarded as a co-ordinate transformation. The walk $(\vec{0}, \vec{\alpha}_1)$ thus reads $(\vec{\beta} = \hat{\alpha}_1(\vec{0}), \vec{0})$ in the new co-ordinates. Now we may append the second step in this coordinate system, say $\vec{\alpha}_2$. After applying the corresponding transformation we have $(\hat{\alpha}_2(\hat{\alpha}_1(\vec{0})), \hat{\alpha}_2(\vec{0}), \vec{0})$. In general we obtain a representation of the first k steps of w in the k-times transformed co-ordinate system by applying the coordinate transformation $\hat{\alpha}_k$ associated with the step $\vec{\alpha}_k$ to the previous representation and appending $\vec{0}$. Thus the coordinates of the k-th point of the walk can be written in the form

$$\vec{x}_k = \hat{\alpha}_1^{-1}(\hat{\alpha}_2^{-1}(\hat{\alpha}_3^{-1}(\ldots \hat{\alpha}_{k-1}^{-1}(\hat{\alpha}_k^{-1}(\vec{0}))\ldots))) \stackrel{\text{def}}{=} \Psi_k^{-1}(\vec{0}) \tag{6}$$

Ψ_k is the affine transformation that takes the original coordinate system into the k-times transformed system after the k-th step of the walk. By construction we have $\Psi_k(\vec{x}_k) = \vec{0}$, and thus

$$\Psi_k(\vec{x}) = \mathbf{T}_k(\vec{x} - \vec{x}_k) \tag{7}$$

where $\mathbf{T}_k$ is a linear transformation. Of course we have $\mathbf{T}_k = \mathbf{S}_k\mathbf{T}_{k-1}$. Now consider the k-th step of the walk itself; we have

$$s_k = \vec{x}_k - \vec{x}_{k-1} = \Psi_{k-1}^{-1}(\hat{\alpha}_k^{-1}(\vec{0})) - \Psi_{k-1}^{-1}(\vec{0}) = \mathbf{T}_{k-1}^{-1}\vec{\alpha}_k - \vec{0} = \mathbf{T}_{k-1}^{-1}\vec{\alpha}_k \tag{8}$$

We also know that the $(k-2)$nd point in the walk has the coordinates $\vec{\beta}$ in the co-ordinate system defined by Ψ_{k-1}. Thus $\Psi_{k-1}(\vec{x}_{k-2} - \vec{x}_{k-1}) = -\mathbf{T}_{k-1}s_{k-1} = \vec{\beta}$, and

	B	F	L	R
b	f	b	r	l
f	b	f	l	r
l	r	l	b	f
r	l	r	f	b

	B	L	R
b	f	d	l
f	b	l	d
l	d	u	r
r	u	f	b
u	r	b	f
d	l	r	u

Figure 7: Move tables for the square lattice `SQ` (left) and the honeycomb lattice `HEX` (right). The lattices themselves are shown in figure 6. Relative moves are given in capitals, lower case letters refer to absolute moves. Note that the honeycomb lattice is a subset of the triangular lattice`TRI` , which makes use only of the relative moves B, L, and R, while F, U, and D do not occur.

hence $s_{k-1} = -\mathbf{T}_{k-1}^{-1}\vec{\beta}$. Thus we can easily recover the coordinates $\vec{x}_k$ provided the individual transformation $\mathbf{S}_k$ are known.

The ambiguity in the assignment of the affine automorphism $\hat{\alpha}_k$ to the moves $\vec{\alpha}_k$ can be removed by requiring that s_k be determined by s_{k-1} and $\vec{\alpha}_k$ alone. This amounts to determining the linear transformation $\mathbf{S}_k$ depending on the absolute move s_{k-1} of the previous step and the current choice of the neighbor $\vec{\alpha}_k$. The advantage of this procedure is that it not necessary to explicitly determine the linear transformations at all. Let $\mathcal{D}$ denote the set of all possible differences between consecutive steps, $\mathcal{D} = \left\{ s = \vec{y} - \vec{x} \,\middle|\, \vec{x} \in \mathcal{L} \text{ and } \vec{y} \in \mathcal{N}(\vec{x}) \right\}$. We shall call $\mathcal{D}$ the set of *absolute moves* in $\mathcal{L}$. All we really need for handling the walks then is a table containing the assignments

$$\mathcal{D} \times \mathcal{N}(\vec{0}) \mapsto \mathcal{D} : \quad (s_{k-1}, \vec{\alpha}_k) \to s_k \tag{9}$$

This construction is illustrated in figure 7 for the square lattice and the honeycomb (hexagonal) lattice. We shall adopt the convention to use lower case letters for the possible choices of absolute moves s_k and capitals for the "relative" moves $\vec{\alpha}_k$.

A walk on $\mathcal{L}$ can thus be encoded as the string of the letters representing the "relative moves". The necessary alphabet size is z, the vertex degree of the lattice graph. As we claimed self-avoidingness of the walks we only need $z - 1$ letters (this generally applies to walks that never reverse a step)[6]. In order to

[6]A walk that does not reverse a step uses only the relative moves associated $\mathcal{N}(\vec{0}) \setminus \{\beta\}$

reconstruct the coordinates $\vec{x}_k$ of the individual steps of the walk we iteratively translate the relative moves into the absolute increments using tables as the ones in figure 7. Practically these computations are performed in terms of the integer lattice coordinates rather than the "actual" coordinates of $\vec{x} \in \mathbb{R}^m$.

Usage of relative moves is well established [90], our implementation has been adapted to apply to any regular lattice. For a more thorough survey on the work that has been done with lattice biopolymers in our group in the 1990ies see [129]. The main advantage of our approach is that it becomes very easy to handle walks on different lattices and with different dimension within the same computer program as the move-tables described in equation 9 and figure 7 can be implemented as simple look-up tables of characters without any reference to a coordinate system. The algorithm has several advantages over representing structures by absolute moves or integer coordinates that turn out to be very useful:

- Lattice independent programming of folding algorithms and structure comparison is possible
- Concatenation of strings corresponds to elongation of the first walk
- Storage requirements are kept small (compression of walk-data strings)
- Structure comparison is achieved by simple string comparison methods: Hamming distance and sequence alignment define a metric distance measure in shape space
- Simple point mutations, i.e., the exchange of one relative move within the walk by another one correspond to pivot moves [104]

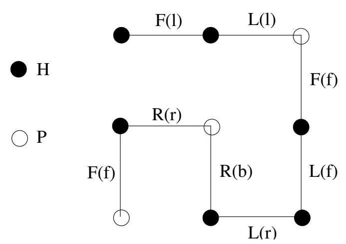
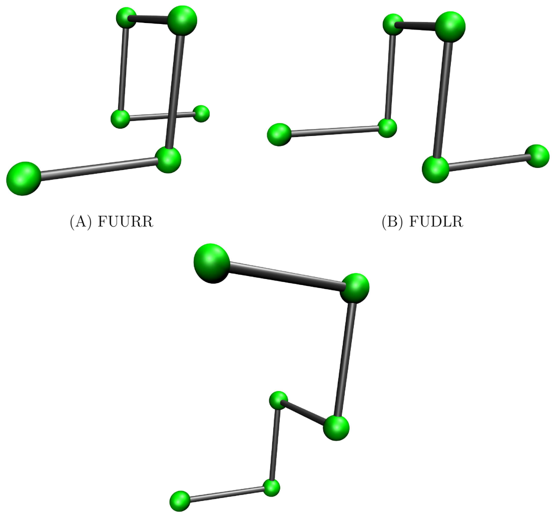


Figure 8: An example for the relative encoding of a SAW, FRRLLFLF. Absolute directions are shown in parentheses. The labels **P** and **H** refer to Dill's **HP** lattice heteropolymer model, see section 4 for details.

The drawback of this approach is that strings of relative moves, and therefore self-avoiding walks within our framework, that are subject to point mutations are *not* necessarily enantiomeres in $d > 2$. In other words, pivot moves, in our implementation, do not necessarily yield chiral structures in $d > 2$ since not only rotations, but also reflections are allowed. Mutations (and thus pivot moves) correspond to automorphisms, mapping the lattice to itself [5]. See figure 9 for illustration of this fact. If we wanted to consider chirality correctly, we had to change more than one relative directions within our framework, choose a different move-set or investigate relative move strings with respect to exact automorphsm groups (see [5] for details). These considerations are subject to our current investigations.



(A) FUURR (B) FUDLR

(C) FUDRR

Figure 9: Point mutation of the relative-move string (shown below the structures) in the `SC` lattice. A point mutation does not necessarily yield two enantiomere structures. Assume structure (A) is the start-structure. Point mutation of the third relative direction from U to D yields the structure shown in (C). This operation was achieved by reflection **and** rotation. If we wanted to get reflection (B) of the start structure, we had to change *two* relative moves within the self-avoiding walk.

What still needs to be established is the total number of SAWs of given length, i.e. the size of the conformation space of a lattice protein. Let $\mathcal{L}$ be some regular d-dimensional lattice and ω an N-step self-avoiding walk $w = (\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots \vec{x}_N)$ of distinct lattice points in $\mathcal{L}$ such that each point is a nearest neighbor of its predecessor (section 2.6). We shall restrict attention to the simple (hyper-)cubic lattice $\mathbb{Z}^d$ and assume that all walks begin at the origin ($\omega_0 = \vec{0}$). The number of N-step SAWs on $\mathbb{Z}^d$ starting at the origin and ending anywhere, c_N, are believed to have asymptotic behavior [141]:

$$c_N \sim \mu^N \cdot N^{\gamma-1} \tag{10}$$

as $N \to \infty$. μ is called the *connective constant* (or *effective coordination number*) of the lattice, γ is the *critical exponent*. The connective constant is definitely lattice-dependent, while the critical exponents are believed to be universal among lattices of a given dimension d. See the table below for some known numerical vales of μ and γ.

dim	Lattice Type	μ	ref.	γ	ref.
2	SQ	2.63820	[8, 44, 64]	1.34275	[64]
	TRI	4.15076	[64, 65, 66]	1.343	[64]
	HEX	1.84777	[64]	1.345	[64]
3	SC	4.68391	[64]	1.161	[64]
	BCC	6.53036	[27, 125]	1.161	[17]
	FCC	10.0364	[64]	1.162	[64]

Table 1: Asymptotic enumeration of SAWs.

Figure 10 shows results from exhaustive enumerations of SAWs on various lattices. The lengths given in figure 10 represent the maximum length of SAWs that can be generated exhaustively for the respective lattice on a modern workstation (see also section 2.8).

2.7 Potential Functions

As mentioned in the previous section, we use *relative* moves for storing strings and comparing structures. The structure is represented as a self avoiding walk on a regular lattice and the movement of the chain is represented as a sequence of moves where each is encoded relative to the prior.

The energy function for a sequence with n residues $\mathfrak{S} = (\mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_n)$ with $\mathfrak{s}_i \in \mathcal{A} = \{a_1, a_2, \ldots, a_b\}$, the alphabet of b residues and an overall configuration $X = (x_1, x_2, \ldots, x_n)$ on a lattice $\mathcal{L}$ cam be written as the sum of all pairwise

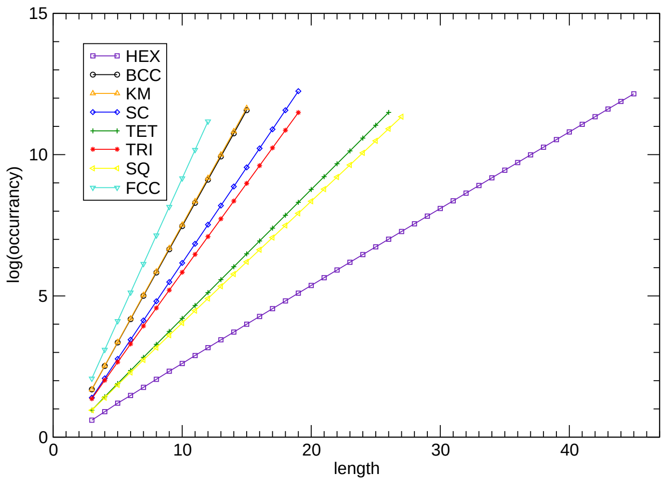


Figure 10: Exhaustive enumeration of SAWs on selected lattices

inter-residue interactions

$$E(\mathfrak{S}, X) = \sum_{i=1}^{n} \sum_{j>i+1}^{n} \mathcal{E}(\mathfrak{s}_i, \mathfrak{s}_j) d_{ij}^{\alpha} f(\mathfrak{s}_i, \mathfrak{s}_j, |i-j|) \tag{11}$$

where $d_{ij} = ||x_i - x_j||$ is the Euclidean distance, $\mathcal{E}_{ij} = \mathcal{E}(\mathfrak{s}_i, \mathfrak{s}_j)$ a pair-potential retrieved from an energy matrix (see below). In our implementation, contributions are considered up to a certain cutoff distance: $d_{ij}^{\alpha} = 0$ if $d_{ij} > d_{max}$. Generally, α is -1 and $d_{max} = 1$ (for compliance with Dill's model). Finally, f respects the dependency of distance within the sequence and takes on 1 in all our calculations.

We implemented two different potentials: **HP** and **HPNX**. Within the "classical" **HP** model [35, 90], a crude simplification is introduced by reducing the various inter-atomic forces to one inter-residue force, the *hydrophobic force*. This unspecific force is assumed to be the dominant contribution to stability and therefore to a large extent determines the 3D structure of the backbone. Heteropolymers are composed from a two-letter alphabet $\mathcal{A} = \{\mathbf{H}, \mathbf{P}\}$ where there is only one stabilizing interaction if, and only if hydrophobic residues (**H**) are neighbors on the lattice but not along the chain. Polar residues (**P**) do not explicitly contribute to the overall energy. Although this model is a crude abstraction, several salient features of real protein structures are implicitly considered: The hydrophobic effect comprises solvent-driven collapse to a native state, chains have (relatively) much conformational freedom and the self-avoiding walk constraint accounts for excluded volume restrictions [39].

The **HPNX** model is a generic extension of the **HP** model and mimics "electrostatic" interactions between negatively charged residues (**N**) and those with a positive charge (**P**) as well as repulsions within these classes. A third class of apolar residues is "neutral" (**X**). As the frequency of **H**s - within a random distribution - is the same as in the **HP** model, the **HX** subset corresponds exactly to the **HP** model [128].

The **HP′** and **YhHX** models shall also be mentioned here for the sake of completeness, the first being derived from the conventional **HP** potential and including a stronger overall attracting force. The latter is a modified form of Crippen's empirical potential [31] which consists of four different classes of residues. Appendix A lists associated energy matrices.

At this point it seems fair to consider a simple question: Is it correct to model proteins with only 2 (resp. 4) different types of monomers, or is this simplification too crude? We could also formulate this question as: What is the minimum number of different monomers to fold a functional protein? Experimental studies have shown that the full sequence complexity of naturally occurring proteins is not necessarily required to design a functional, rapidly folding protein. In fact, proteins with a drastically reduced set of amino acids (compared to the 20 naturally occurring ones) have been successfully designed experimentally in the last years (e.g. [133]). Some amino acid residues have similar physico-chemical properties and their substitutions are tolerated in many regions of a protein sequence.

Govindarajan and Goldstein proposed that evolutionary pressure is responsible for a protein to fold fast. Studying the *foldability* of structures in a lattice model, they suggested that structures with larger optimal foldability should tolerate more sequences and be more robust to mutations [58, 59]. Within this context, we can also speak of the *designability* of a structure, that is the number of sequences that have that structure as their unique lowest-energy state. Li et al. studied the designability of all compact structures in **HP** lattice models of sizes $3 \times 3 \times 3$ (`SC` lattice) and 6×6 (`SQ` lattice) [97]. They found that structures differ drastically in their designabilities and that a small number of structures emerge with designabilities much larger than the average. In another contribution, Li et al. recently calculated designabilities with all 20 amino acids with empirically determined interaction potentials and found that the designability of a structure is not sensitive to the alphabet size as long as hydrophobic interaction is included in the potential [98].

An interesting contribution was given recently by Fan and Wang, who published rigorous investigations with reduced alphabet sizes. According to them, the lower bound of amino acid types required for a protein to fold into a stable structure is around ten [45].

2.8 Lattice Protein Folding Algorithms

One of the main intentions in lattice protein studies has always been the search for a protein's ground state only with knowledge of its sequence. Furthermore, it is not clear whether a ground state - if found - is unique. As illustrated in section 2.4, efficient algorithms to determine the ground state and all suboptimal structures within a predefined energy interval exist at least for RNA. In contrast to that, the structure prediction problem for lattice proteins was shown to be NP-complete, even for the **HP** model [150]. Crescenzi et al. [30] gave a proof for the two-dimensional case, the three-dimensional case was proved by Berger et al. [9]. At present no polynomial time algorithms are known for an NP-complete problem and it is generally believed that such an algorithm does not exist [52]. This imposes a major drawback in the ability to investigate protein properties within reasonable time scales and computer resources. Nevertheless, a large variety of approximation algorithms has been proposed so far. A resource intensive genetic algorithm based on Monte Carlo techniques in the square lattice yields good results for fairly long chains up to a length of 60 monomers [151]. Another approach tries to approximate the whole density of states by recursively counting up low energy states [144]

Bornberg-Bauer and Renner provided a fast, straightforward heuristic algorithm for **HP**-type lattice proteins [10]. Their deterministic "greedy chain-growth algorithm" is designed after the concept of unguided, cotranslational folding of a nascent peptide and runs in $\mathcal{O}(n)$ in chain length. A "frozen" start structure is taken as starting point and all possible combinations (of a predefined window-size) of subsequent lattice positions is evaluated for the energetically "best" structure (i.e. the structure with maximum hydrophobic interactions). The first relative move of the so calculated "current" optimal structure is then appended to the end of the frozen part of the SAW and the procedure is repeated until an optimal structure of given length is reached. The method is fast enough for statistical investigations on large ensembles of structures.

Several interesting contributions were given by Backofen and Will [2, 4, 5] within the last years. Inspired by findings of Yue and Dill [165] (who proposed a branch-and-bound algorithm for finding optimal structures on the `SQ` lattice), they designed a global optimization technique for **HP**-kind lattice models on the cubic (`SC`) and the (more demanding) face-centered cubic (`FCC`) lattice. The method is based on constraint optimization and can successfully fold sequences up to length 300. In addition, the method has been applied to enumerate all minimum energy conformations for sequences up to length 48.

When talking about folding algorithms for lattice heteropolymers one must bear in mind that proteins, contrary to their reputation, do *not* always fold efficiently and spontaneously. Several proteins need some sort of help, e.g. by "chaperon" molecules, others are folded erroneously and are recycled by proteolytic enzymes. A protein's native state is not necessarily the state of lowest free energy [72]. In fact, a biological molecule doesn't have to be absolutely stable, it only has to be functional enough to do its job. Considering this, it seems fair to claim that perhaps the appropriate model for protein folding is an approximation algorithm that is guaranteed to quickly find a near-optimal structure. Hart and Istrail presented an algorithm for the **HP** model that guarantees structures with energy better than 3/8 of the optimal structure [70], although accuracy depends strongly on the lattice and energy function.

We decided to take a different approach in calculating protein structures: Rather than sampling structures by a Monte-Carlo method, we implemented a "brute force" tool (`latticeSub`) to exhaustively generate all SAWs of given length on any of the lattices given in figure 6. Evidently, this approach is restricted to very short chain lengths due to NP-completeness of the problem (see figure 10).

Although chain lengths are restricted to small values, the set of *all* configurations of given length allows for calculation of energy landscapes and thus gives valuable insight into the folding dynamics of short lattice heteropolymers. A different approach to calculate only a low-energy fraction of the state space will be given in chapter 6.

3 Biopolymer Folding - Energy Landscapes

In order to understand the (folding) dynamics of biomolecules, we need to investigate the underlying *energy landscape*, i. e. we are interested in the topology of the *folding landscape*. The folding landscape (or Potential Energy Surface, PES) of a biopolymer molecule is a complex surface of the (free) energy versus the conformational degrees of freedom. In formal terms, we need three things to construct an energy landscape [127]:

- A set X of configurations
- a notion $\mathfrak{M}$ of neighborhood, nearness, distance or accessibility on X, and
- an energy function $f : X \rightarrow \mathbf{R}$

In this chapter we will work out the details of non-degenerate energy landscapes in terms of a theoretical description, a thorough investigation for degenerate landscapes will be given in the next chapter.

3.1 The Move Set

The *conformation space* X of a biopolymer sequence is the total set of configurations S compatible with this sequence. In the RNA case, the set of configurations is given by the secondary structures which are compatible with a particular sequence. In the lattice protein case, all self-avoiding walk structures of a given length are considered as allowed conformations. The degrees of freedom are the allowed transformations provided by a *move set* $\mathfrak{M}$.

Depending on the coarse-graining of the energy, conformation space (and hence the associated energy landscape) can be highly degenerate, especially in the lattice protein case (sections 4.1 and 7.3.2). *A priori* it is not clear how to move in such a complex space, therefore a set of rules is needed to control the movement. Such a set of rules is called a **move set**. It is basically a collection of operations, which, applied to an element of X, transforms this element into another element of X. Strictly spoken a move set is an order relation on X, defining *adjacency* between the elements of X. It fixes the possible conformational changes that can take place in a single step during the simulation of folding and thus defines the topology of the conformational space. The following properties are important for move sets:

1. Each move has an inverse counterpart. At thermodynamic equilibrium the quotient of forward and backward reaction rates must give the microscopic equilibrium constant (If there is no backward reaction, the law of microscopic reversibility is broken).

2. The outcome of an operation always leads to an element of the underlying state space (Any operation yielding an element outside the state space is illegal).

3. The move set has to be ergodic[7]. This means, starting from an arbitrary point of the state space every other point must be reachable by a sequence of legal operations.

4. Every move set defines a metric on the underlying state space.

Two more terms are relevant for the further discussion. A *trajectory* is defined as a sequence of consecutive states of the state space generated by a series of legal operations from some initial state. A *path* (or *folding path*) is defined as a cycle free trajectory, more concrete, each state occurs only once within the sequence of adjacent states. Any trajectory can be transformed into a path by eliminating the cycles.

3.1.1 Move Set: RNA

The most elementary move set, on the level of RNA secondary structures consists of insertion and deletion of a single base pair (i, j) in agreement with the knot-free restriction (section 2.4). It is always possible to construct a path between any two $S_i, S_j \in X$ by using operations from this move set. To find such a path, remove from S_i all base pairs that do not occur in S_j, and insert afterwards into this intermediate structure S_k all base pairs from S_j that do not occur in S_i. (Note, that $S_k = S_i \cap S_j$ can be the empty set, which resembles the open chain, being as well an element of X). Since every element of X can be connected to every other element of X by a path, it follows that this is an ergodic move set

[7]Although aware of this fact we implemented a **non-ergodic** move set for lattice proteins, see section 3.1.2

on X. The highly cooperative *zippering mechanism* observed in helix-formation events of nucleic acids can be properly described with this move set.

Although the simple move set is sufficient in principle, it lacks efficiency in practice. Hence, a third move consisting of a base pair shift must be introduced (figure 11). The so called *shift move* converts an existing base pair (i, j) into a new base pair (i, k) or (l, j) in a single step. See ref [47] for details on the implemented move sets for RNA.

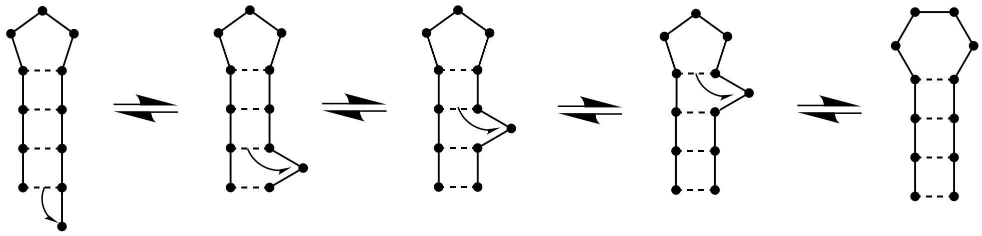


Figure 11: Formation of helices with incomplete base pairing. The effect that anneals intermediately mismatched helices is called *defect diffusion*: The bulge can easily migrate along the helix. For the left to right transformation the shift moves are indicated by arrows

3.1.2 Move Set: Lattice Proteins

pWithin the framework of SAWs, one can classify elementary moves according to several properties, i.e. it is necessary to determine whether a move is

- local or non-local,
- N-conserving or N-changing,
- endpoint-conserving or endpoint-changing,

A **local** move is one that alters only a few consecutive sites (or beads) of a SAW, leaving the other sites unchanged. In other words a local move 'cuts' a small piece from the original SAW and splices in a new local conformation. We say that a move is k**-local** if each move affects at most k consecutive positions in the SAW. Of course, it is always necessary to check if the proposed new walk is still self-avoiding. A **non-local** move, in contrast, alters a large fraction of the SAW. Since a non-local move is rather radical, the resulting new walk usually violates the self-avoidance constraint [141]. N**-conserving** moves are those in which the excised and spliced-in subwalks have the same number of beads, whereas a **N-changing** move has the freedom to splice in a piece with different length than the original piece. We will exclusively examine N-conserving moves in the following.

One of the simplest, yet efficient moves is called **pivot move** [89], a non-local, endpoint-changing move. The elementary move is as follows: A site $\vec{x}_k$ on the walk $x \in X$ is chosen as a pivot point and a symmetry operation of the lattice (rotation and/or reflection) is applied to the part of the walk subsequent to the pivot point (namely $\vec{x}_{k+1}, \ldots, \vec{x}_N$), using the pivot point as the origin. The resulting walk is accepted if it is self-avoiding, rejected otherwise. See figure 12 for an illustration of pivot move.

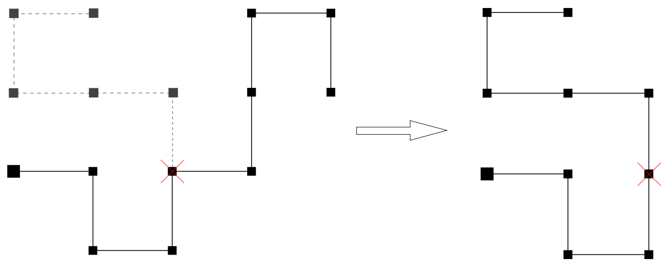


Figure 12: A pivot move in the `SQ` lattice (here a +90° rotation). The pivot site is marked with a red cross, dashed lines indicate the proposed new segment

We have previously mentioned that there are several advantages in considering pivot moves (section 2.6). First, pivot moves provide an *ergodic* move set, that means that any state of the configuration space must be reachable from any other state by a finite set of operations according to a certain move set. Another advantage is that implementation of pivot moves is fairly easy. Within our implementation, point mutations of a SAW in relative move notation yield pivot moves. Originally, the pivot algorithm was presented as a dynamic Monte Carlo algorithm which generates SAWs in a canonical ensemble with one endpoint fixed and the other endpoint free. Ergodicity of pivot moves was proven by Madras and Sokal [104] for the simple (hyper)cubic lattice. Another ergodicity proof was given in [102]. In fact, the same authors proved that every local, N-conserving Monte Carlo algorithm is *non-ergodic* for sufficiently large N [103]. In other words, some SAWs cannot be transformed into some others by any sequence of allowed moves.

Despite of this fact we implemented another set of elementary that belong into the class of *non-ergodic* local, N-conserving moves: End-, corner-, and crankshaft moves (figure 13). We did this in accordance with literature (see e.g. [140]) to investigate differences in folding pathways and study the influence of the chosen move set with respect to the topology of the whole energy landscape. One could

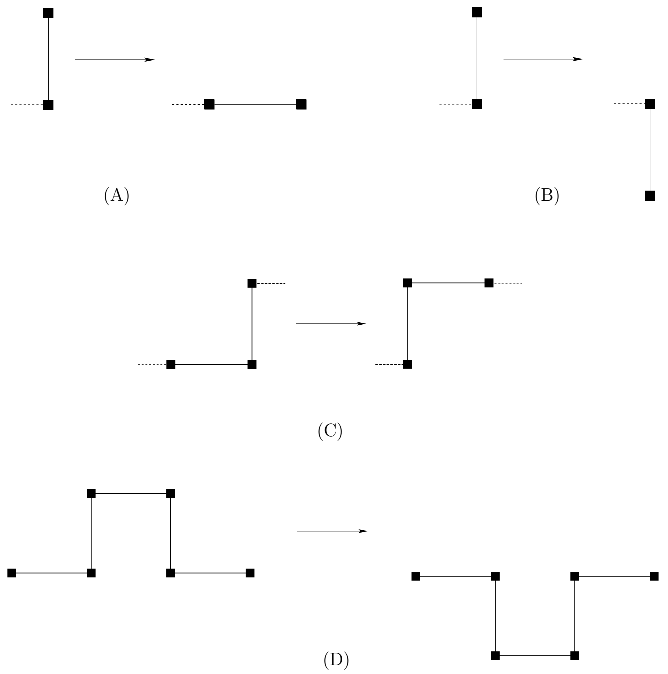


Figure 13: Crankshaft-, corner- and end moves. (A) and (B) show the possible end moves. Note that move (B) allows for a 180° flip within *one* elementary move. (C) shows a corner move, the only possible one-bead internal move. (D) shows a 180° crankshaft move.

of course ask whether the non-ergodicity of these moves will be a problem. The answer to that depends on the type of information one is seeking. If one is seeking moderately accurate numerical data for modest values of N, then perhaps the exclusion of some configurations causes only a small error. We are interested in protein folding kinetics and as long as the native state is accessible the existence of a small fraction of inaccessible states is negligible. Just in the same way as highly unlikely states are irrelevant for real proteins, we argue that these states are negligible for our purposes. However, the fraction of configurations that are inaccessible from the open chain configuration is subject to future investigations. Figure 14 shows examples of SAWs on the `SQ` and `SC` lattice that are inaccessible, starting from an open chain, by moves shown in figure 13. An interesting aspect are knotted configurations. It is known that real proteins do not have tight knots and hence it is fairly unlikely that these configurations are 'visited' during a folding event since a protein would have to escape such a configuration after being knotted once.

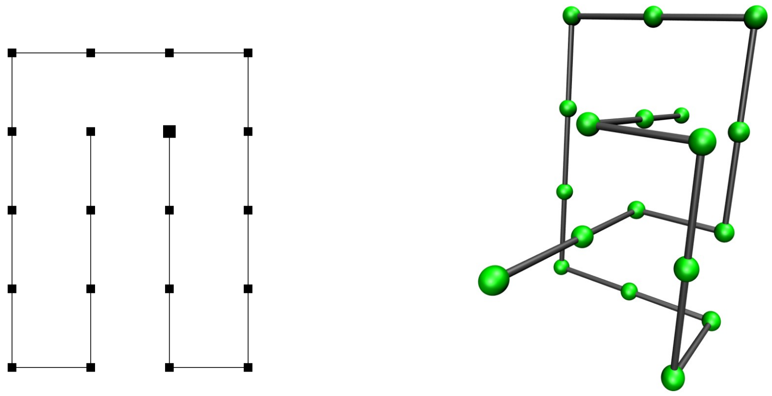


Figure 14: Non-ergodicity of local moves. The configuration on the left (SC lattice "double cul-de-sac", ref [104]) is frozen and not accessible by any moves from figure 13. The knotted conformation on the right (SC lattice) can not be unknotted by the moves shown in figure 13. Hence it is impossible to reach this conformation using the same moves. As long as conformations like this are not the native state they will pose no problem.

3.2 Energy Landscapes: Mathematical Definitions

Within the framework of the folding landscape we can meaningfully speak of local minima or metastable states, their basins of attraction, and the saddle points separating them. Formally[8], a structure $x \in X$ is a local minimum of E if $E(x) \leq E(y)$ for all its neighbors, $(x, y) \in \mathfrak{M}$. A gradient walk is defined as follows: starting from $x \in X$ we move to its neighbor y with minimal energy if $E(y) < E(x)$. If the minimum energy neighbor y of x is not uniquely defined we use a deterministic rule to break the tie, for instance, by choosing the structure that comes lexicographically first. The step from x to $y = \gamma(x)$ is repeated until we reach a local minimum where the walk terminates, $\gamma(x) = x$. The local minima are therefore the attractors of the map $\gamma : X \to X$ and each $x \in X$ is mapped to a unique local minimum $z = \gamma^\infty(x) = \gamma^t(x)$ by a finite number t of applications of γ. The basin of attraction of a local minimum z, $\mathcal{B}(z)$, consists of all structures that are mapped to it by the gradient walk, i.e. $\mathcal{B}(z) = \{x \in X | \gamma^\infty(x) = z\}$.

[8]Although we labeled structures with S up to this point, we switch the notation to x here to stress that he following definitions are generic and not specific for RNA or lattice proteins.

Below we will need the (trivial) fact that these "gradient basins" of the local minima form a partition of X.

Let us now turn to the transitions between local minima. The energy of the lowest saddle point separating two local minima x and y is

$$E[x,y] = \min_{\mathbf{p}\in\mathbb{P}_{xy}} \max_{z\in\mathbf{p}} E(z) \tag{12}$$

where $\mathbb{P}_{xy}$ is the set of all paths $\mathbf{p}$ connecting x and y by a series of subsequent moves. The saddle-point energy $E[\,.\,,\,.\,]$ is an ultra-metric distance measure on the set of local minima.

In the simplest case the energy function is non-degenerate, i.e., $f(x) = f(y)$ implies $x = y$. Then there is a unique saddle point $s = s(x,y)$ connecting x and y characterized by $E(s) = E[x,y]$. This definition of a saddle point is more restrictive than in differential geometry where saddles are not required to separate local optima. For each saddle point s there exists a unique collection of configurations $\mathcal{V}(s)$ that can be reached from s by a path along which the energy never exceeds $E(s)$. In other words, the configurations in $\mathcal{V}(s)$ are mutually connected by paths that never go higher than $E(s)$. This property warrants to call $\mathcal{V}(s)$ the *valley below the saddle* s. Furthermore, suppose that $E(s) < E(s')$. Then there are two possibilities: if $s \in \mathcal{V}(s')$ then $\mathcal{V}(s) \subseteq \mathcal{V}(s')$, i.e., the valley of s is a "sub-valley" of $\mathcal{V}(s')$, or $s \notin \mathcal{V}(s')$ in which case $\mathcal{V}(s) \cap \mathcal{V}(s') = \emptyset$, i.e., the valleys are disjoint. This property arranges the local minima and the saddle points in a unique hierarchical structure which is conveniently represented as a tree, termed *barrier tree* (see Fig. 15). Since saddle points separate local optima, each valley $\mathcal{V}(s)$ contains (in the non-degenerate case at least two) local minima $z_1, \ldots, z_k$. Conversely, $\mathcal{V}(s) \subseteq \bigcup_k \mathcal{B}(z_k)$, i.e., the valley of s is contained in the union of the basins of attraction of the metastable states "below" s. The metastable states therefore form the tips (or leafs) of a tree. This so-called barrier tree is the subject of the next chapter. In the case of degenerate landscapes an analogous construction is possible when certain saddle points with the same energy are collected into equivalence classes, as we will see in section 4.3.

4 Barrier trees

We set up the requirements to investigate energy landscapes of biopolymers in the last chapter. What we have at hand at this point is a straightforward decomposition of non-degenerate landscapes into basins surrounding local minima, connected by saddle points. The decomposition of landscapes into basins and investigation of trees that represent local minima and their connecting saddle points has been developed independently in different contexts, among them $\pm J$ spin models [86, 87], potential energy surfaces (PES) for protein folding [7, 53] and molecular clusters [41, 155] as well as the kinetics of RNA folding [47] (see section 7.1). This chapter is dedicated to a more thorough investigation of barrier trees. We will give examples of barrier trees for RNA and lattice proteins, present an algorithm for efficient computation of such barrier trees and investigate a rigorous concept of barrier trees for degenerate landscapes.

4.1 Examples

Barrier trees of RNAs are usually non-degenerate and the straightforward definitions from section 3.2 can generally be applied "as-is" to RNA energy landscapes. The situation is quite different with lattice proteins, since energy landscapes and thus barrier trees usually exhibit a large degree of degeneracy here. This is evident since the model of lattice proteins implies rigorous assumptions, i. e. fixed bond lengths- and angles and an alphabet-dependent energy function (section 2.7). In order to illustrate the difference, we give the tree representation of the lowest 10 minima in the (non-degenerate) energy landscape of a random RNA sequence with length $n = 40$ in figure 15. Leaves 1-10 correspond to the valleys of the landscape, while saddle points (labeled with capital letters for A to I) are displayed by internal nodes. Saddle point energies can be read off easily, the scale on the left indicates values in kcal/mol. The energy of barrier 3, for example, is $B(3) = E(A) - E(3)$, whereas the Energy barrier to reach 1 (i.e. any structure from the right subtree below saddle E) from 3 is $E(3 \to 1) = B(3) + (E(C) - E(A)) + (E(D) - E(C)) + (E(E) - E(D)) = 1.0 + 1.1 + 0.6 + 0.38 = 3.08$ kcal/mol ($T = 310.15K$). Figure 17, in contrast, shows barrier trees of degenerate landscapes.

The fundamental question concerning these energy landscapes still is: What influ-

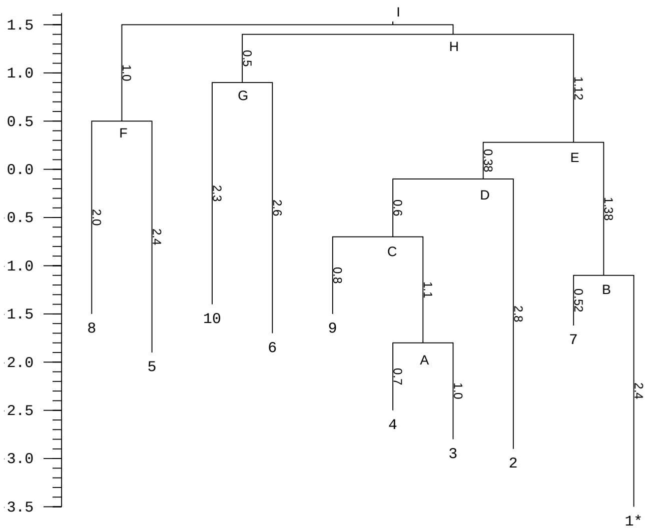


Figure 15: A typical barrier tree of a short random RNA sequence `UUGGACCCAUUC-GAUCCCAGACCUUCAAGGCUUCUCUGUC` with length $n = 40$ as calculated with `barriers`.

ences the ruggedness? In fact, the *definition* of neighborhood strongly influences this feature of the surface. In other words the choice of the move set critically forms the topology of the energy landscape. This is shown in figure 16 for RNA. As illustrated in section 3.1.1, we have two move sets available: insertion/deletion (left plot) and insertion/deletion/shift (right plot). Since the first one is a subset of the second, all local optima of the latter are also local optima of the simple move set. The local minima are again labeled in ascending order starting with the ground state. Equivalent minima are labeled identically in both trees, corresponding local minima are given in brackets in the left plot. Local minimum 8 occurs only within the simple move set (left), whereas local minimum 20 in the right plot just occurs here because is has not been seen yet in the left plot, i.e. it has a higher energy up to which the algorithm couldn't get in the left plot.

Within the framework of lattice proteins and SAWs the situation is different. The simplest lattice is of course the square lattice `SQ`, which was originally used by Lau, Chan and Dill [24, 35, 90] to establish their model. Figure 17 shows two barrier trees for a random **HP**-sequence `HHHPHHPHHPHHHHPH` with $n = 16$ on this lattice. The most noticeable difference (compared to RNA energy landscapes) is definitely the high degree of degeneracy. We only show the deepest 100 local minima of the energy surface. There are two degenerate ground states ($E = -9$), 23 local minima with $E = -8$ and the rest is energetically indistinguishable with

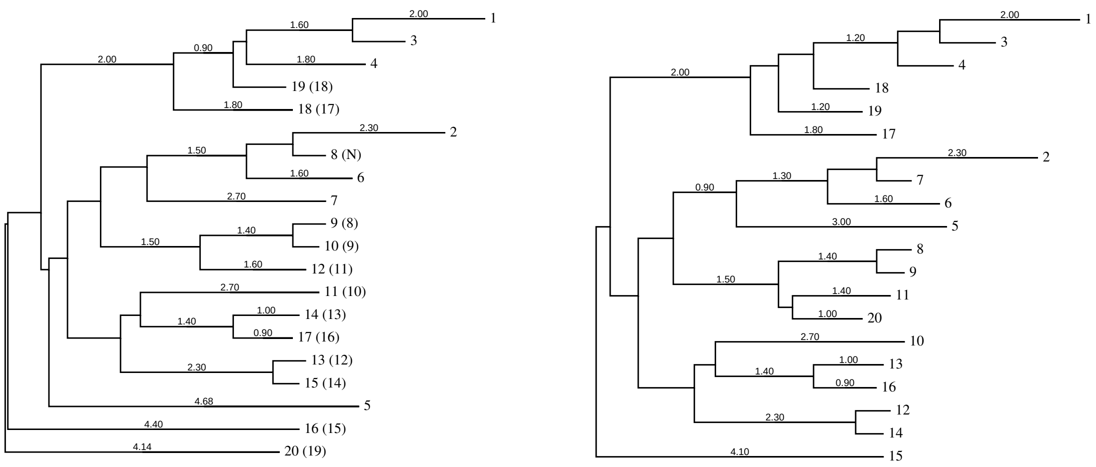


Figure 16: The tree representation of the energy landscape of a typical RNA sequence. The 20 lowest local minima are shown for the simple move set (left plot: insertion/deletion) and the enhanced one (right plot: insertion/deletion/shift).

$E = -7$. Degeneracy can easily be explained as an artefact of the underlying model, i. e. of the underlying energy function. As we are dealing with a two-letter alphabet, the energy of a certain structure is given by the negative sum of all nearest-neighbor non-bonded `H` monomers (section 2.7 and appendix A). This and the restriction to fixed bond lengths and bond angles sets the stage for highly degenerate barrier trees. If we wanted to circumvent this problem, we could easily switch to a bigger alphabet, e.g. the **HPNX** alphabet. A more realistic model would of course include a larger alphabet (remember the magic number of 10 letters required to fold a functional protein (ref. [45])). Another possibility would be to define more complex potential functions than those described in section 2.7.

As in the RNA case we are interested in the influence of the move set on the general features of the landscape. The upper tree in figure 17 was calculated with pivot moves as elementary move operation, whereas in the lower tree crankshaft-, corner-, and end moves were considered. We mentioned earlier that the latter set of elementary moves is not ergodic. For the upper plot we used an exhaustive search strategy to find all SAWs[9] of length $n = 15$. After eliminating mirror-image structures, we found 802076 SAWs of length $n = 15$. For the lower plot, we used the program `latticeFlooder` (chapter 6) to generate a total of 800829 SAWs starting from an open chain conformation. 1247 SAWs are not accessible

[9]note that a SAW is always one element shorter that the sequence "laid" onto it

by the latter move set starting from the open chain conformation.

Generally, the number of neighbors a structure can attain with a single elementary move is of critical importance for the topology of the energy landscape. If many moves are possible (with respect to self-avoidance), one can expect to get fewer local minima and lower barrier heights. The non-local pivot move allows to convert a single structure to a *completely* different structure in one move, hence energy landscapes that are calculated with pivot moves are generally slightly 'smoother' than others. This can clearly be seen in the upper tree of figure 17, where the maximum barrier height is 5. When we change the move set to crankshaft-, corner- and end moves (this is just possible in the `SQ` and `SC` lattices within our implementation), valleys become 'deeper' and barrier heights are bigger (lower tree in figure 17). The number of neighbors that are reachable in an elementary step is not so large (as we are dealing with a local move set).

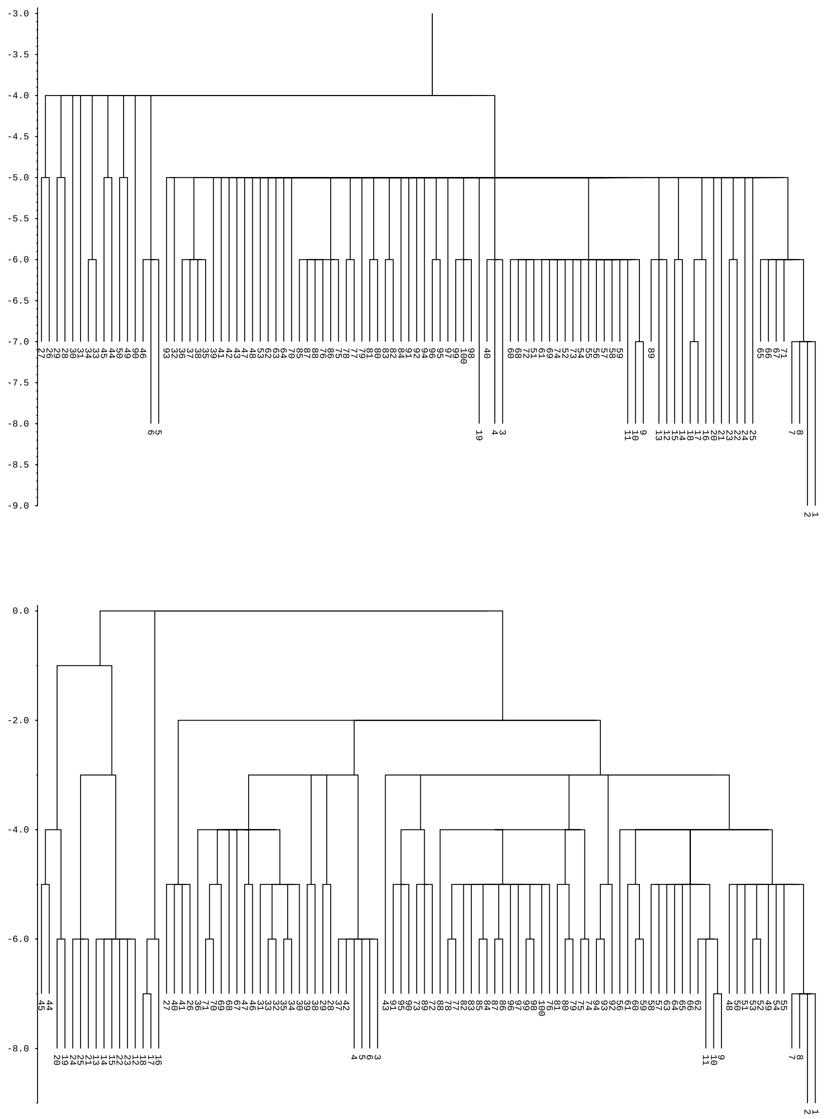


Figure 17: Energy landscapes of a random lattice protein, sequence `HHHPHHPHHPHHHHPH`, with length $n = 16$. Influence of the move set on the topology of the landscape. Pivot moves (upper plot) yield smoother landscapes, local moves (lower plot) yield deeper basins. See text for details.

4.2 The algorithm of `barriers`

After illustrating principal properties of barrier trees, the next step in our investigation of the energy landscapes of biopolymers is to *calculate* the barrier tree of the interesting biomolecule. We use the program package `barriers`[10] to calculate barrier trees. The framework of RNA secondary structures and SAWs allows for a *discrete* description of the problem, which, although computationally challenging and in contrast to continuous models, sets the stage for an exact enumeration of the underlying states. This holds at least for moderate-size state spaces. Although the algorithm was described in previous work [161], we review the main concept here.

`barriers` constructs the barrier tree directly from an energy-sorted list of all configurations, so a prerequisite step is to generate these lists of "suboptimal" configurations. `RNAsubopt`, which computes all secondary structures below a certain energy threshold is used in the RNA case. In the lattice protein case we have two tools at hand: `latticeSub` (section 2.8), yielding exhaustive enumeration of SAWs on a given lattice up to a certain (moderate) length (figure 10) and `latticeFlooder` (see chapter 6) that starts from a given SAW and generates all neighbor structures according to a selected move set. These neighbor structures are again considered start structures and this procedure is continued until a) all structures up to a predefined energy threshold or b) a predefined amount of structures are found.

Algorithm 1 lists pseudo-code for the main routine of `barriers`. During the calculation, two lists of local minima are needed, each of which are required to be empty at the start of the algorithm. $\mathcal{B}$ is a *global* list of all local minima found in the landscape, $\mathcal{K}$ is a *local* list of local minima that contains neighbors of the current structure x. As mentioned before all suboptimal structures are processed in energy-ascending order. Starting with a structure x, the first step is to generate its complete neighborhood $\mathcal{N}$ according to the chosen move set and store all neighbor elements y on a stack (line 3). Then each element of this "neighbor stack" is processed: A routine searches a hash if structure y has already been seen in a previous step of the computation and, if so, counts the number b of local minima that contain legal neighbor structures of x. The set $\mathcal{K}$ of local minima containing neighbors is updated immediately (lines 5 and 6). When all

[10] available from `http://www.tbi.univie.ac.at/~ivo/RNA/Barriers`

Algorithm 1 The algorithm of barriers

Require: subopt

1: $\mathcal{B} \Leftarrow \emptyset$
2: **for all** $x \in$ subopt **do**
3: $\quad \mathcal{K} \Leftarrow \emptyset$
4: $\quad \mathcal{N} \Leftarrow generate\_neighbors(x)$
5: $\quad$ **for all** $y \in \mathcal{N}$ **do**
6: $\quad\quad$ **if** $b \Leftarrow lookup\_hash(y)$ **then**
7: $\quad\quad\quad \mathcal{K} \Leftarrow \mathcal{K} \cup b$
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: $\quad$ **if** $\mathcal{K} = \emptyset$ **then**
11: $\quad\quad \mathcal{B} \Leftarrow \mathcal{B} \cup \{x\}$
12: $\quad$ **end if**
13: $\quad$ **if** $|\mathcal{K}| \geq 2$ **then**
14: $\quad\quad merge\_basins(\mathcal{K})$
15: $\quad$ **end if**
16: $\quad write\_hash(x)$
17: **end for**

elements of the neighbor stack have been processed, there are in principle three possibilities for each element x:

- If there was no adjacent structure resulting from the hash-lookup procedure in line 5, the set $\mathcal{K}$ is empty. x is a new local minimum in that case and the global list of local minima, $\mathcal{B}$ is expanded by x (line 10).
- If a neighboring basin was found then structure x is assumed to be "transient", which means that it belongs to a certain basin of attraction (note that this condition is not listed in algorithm 1).
- If $|\mathcal{K}| \geq 2$ (line 12) then structure x has neighbors in exactly $|\mathcal{K}|$ basins. In other words, x is a *saddle point* connecting all local minima in $\mathcal{K}$. In the barrier tree x becomes an internal node. All states from energetically higher basins in $\mathcal{K}$ are copied to the energetically lowest basin in $\mathcal{K}$ (which, of course, must also be an element of $\mathcal{B}$). Let us denote this instance with: energetically higher valleys in $\mathcal{K}$ are *merged* with the energetically lowest

valley (a.k.a. the father) in $\mathcal{K}$ (line 13). Due to this one can say that from the point of view of a structure with an energy higher than the saddle point x, from this point of the algorithm onwards all elements in $\mathcal{K}$ appear as a single valley that is subdivided only at lower energies.

The final step of the loop is to write structure x into a hash for further lookup (line 15)

The flooding algorithm can be visualized with the following thought experiment (figure 18): Imagine a landscape with only two deep valleys A and B where A is energetically lower than B. Those two local minima are separated by the local optimum T, which is a saddle point. Water rises from bottom to top. In the first step (a), only the deeper valley A will be slightly filled with water. For our algorithm this means that all structures that are either below or exactly at the water surface belong to the local minimum A (all other structures are not accessible by now as we go through an energetically sorted list of configurations in ascending order). As the water still rises we encounter a different situation in step 2 (b). Not only A is filled with water, but also the deepest regions of B. From now on there are more possibilities for the configurations to belong to: Depending on which valley is the nearest (from the point of view of the conformation space), i.e. which local minimum contains structures that are legal neighbors of the actual one, a structure can either be added to A or to B.

Imagine the water rises further. The higher the water surface gets, the more structures are being seen. This means that with every increment (concerning the rise of the water) there are more possibilities for a structure which has not been seen so far to have neighbors in one or more of the valleys. Step 3 (c) shows this situation: The saddle point T has been found and there exists at least one structure which has neighbors in A *and* in B. In other words we can say the two lakes coincide. This is of special importance for the algorithm. As soon as T has been proved to be a saddle point, B is merged with its 'father' A and all structures from B can now be accessed as if they would be legal structures belonging to A. However, the algorithm does not stop here. As illustrated in step 4 (d), the water rises on and only valley A is still accessible. The end of the algorithm is reached as soon as either (a) all structures have been processed or (b) a predefined amount of local minima has been found.

The outcome of this procedure is the following information: There exist two

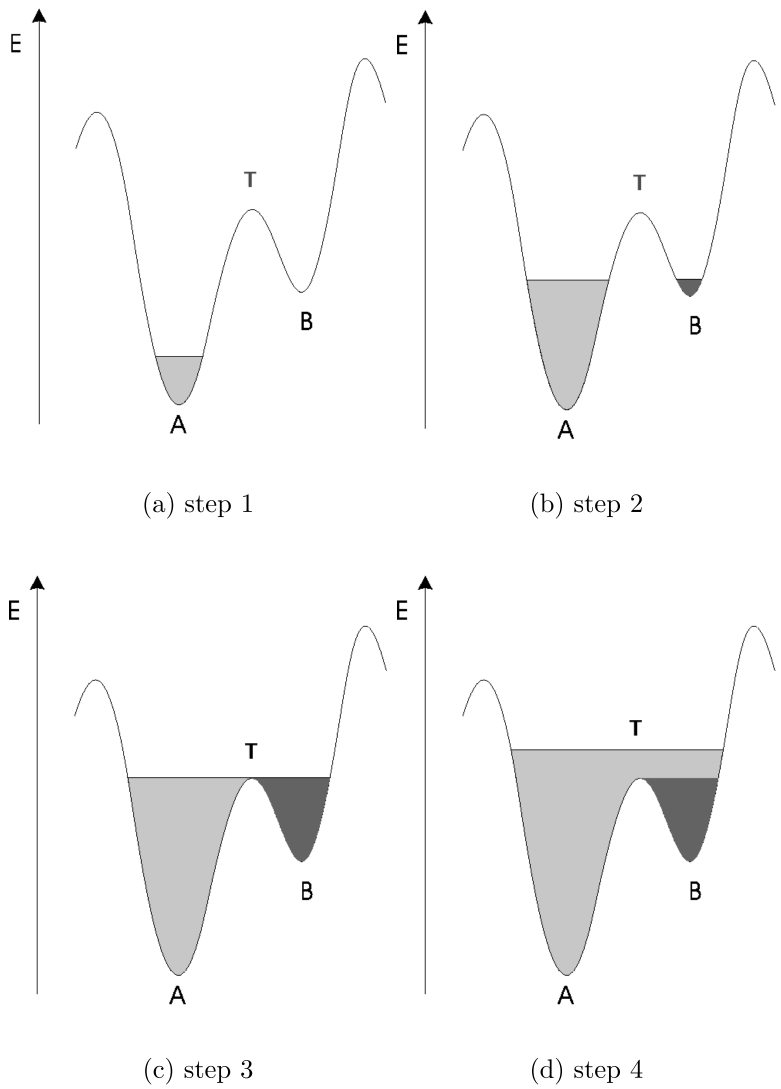


Figure 18: Thought experiment for the flooding algorithm where water rises in a landscape. For details see text.

local minima A and B which are connected by the saddle point T at a certain energy. All structures in A (as B is not accessible any longer) can be neighbors of other structures at higher energy. Evidently this is a very simplified 'gedanken experiment'. Real energy landscapes of biopolymers do not only contain just two local minima, but several thousand.

4.3 Degenerate barrier trees

All definitions for local minima, saddle points, basins and related concepts given so far are readily applicable to non-degenerate landscapes. However, energy landscapes of lattice proteins often exhibit a large degree of degeneracy. In order to treat this type of landscape correctly, additional care has to be taken to find suitable definitions. The problem becomes clear when imagining a flat landscape. Is every point a minimum, or none? If the second alternative is chosen, then the global minimum is not a minimum; clearly one would like to avoid such statements. Two non-adjacent minima should, by intuition, be separated by one or more saddle points. But all points are minima in flat-land, so saddles can be minima as well. We will elucidate such effects and formulate a rigorous concept of barrier trees for degenerate landscapes in this section [49]. Note that the following definitions are given in terms of a connected, undirected, simple graph $G(X, E)$ with vertex set X and edge set E. Within the context of this thesis, single vertices $x \in X$ correspond to structures in the configuration space of a biomolecule.

Notation In the following, we write $\subset$ and $\subseteq$ to distinguish between proper subsets and subsets including the complete set. For the neighbors (adjacent vertices) of $x \in X$ we write

$$\partial x = \partial\{x\} = \big\{y \in X \big| \{x, y\} \in E\big\}. \tag{13}$$

This definitions extend in a natural way to arbitrary vertex sets:

$$\partial A = \big\{y \in X \setminus A \big| \exists x \in A : \{x, y\} \in E\big\} \tag{14}$$

The set ∂A is the *boundary* of A. Furthermore, we write $\overline{A} = A \cup \partial A$ for the graph-theoretic *closure* of a vertex set $A \subseteq X$. The *neighborhood* of x is $\mathcal{N}(x) = \{x\} \cup \partial x = \overline{\{x\}}$.

Definition 1. *A* landscape (G, f) *on a graph* $G(X, E)$ *is a function* $f : X \to \mathbb{R}$*.*

The graph G is often referred to as the *configuration space* in the context of combinatorial landscapes. We write $\min f$ for the value of the global minimum of the fitness function.

Local Minima A vertex x is a local minimum of $f(x) \le f(y)$ for all $y \in \partial x$ (or, equivalently, $y \in \mathcal{N}(x)$); x is a strict local minimum if $f(x) < f(y)$ for all $y \in \partial x$. We write $\mathcal{M}$ for the set of all local minima. Furthermore, let $\mathcal{M}(x)$ be the vertex set of the connected components of $G[\mathcal{M}]$ that contains x. Of course f is constant on $\mathcal{M}(x)$. The set of these components is denoted by $\mathfrak{M} = \{\mathcal{M}(x)|x \in \mathcal{M}\}$.

There are two classes of local minima that can be distinguished by the behavior of the function f on $\partial\mathcal{M}(x)$, see Fig. 19.

(1) $\mathcal{M}(x)$ is a *valley* if $f(y) > f(x)$ for all $y \in \partial\mathcal{M}(x)$.
(2) $\mathcal{M}(x)$ is a *shoulder* if there is a $y \in \partial\mathcal{M}(x)$ such that $f(y) = f(x)$.

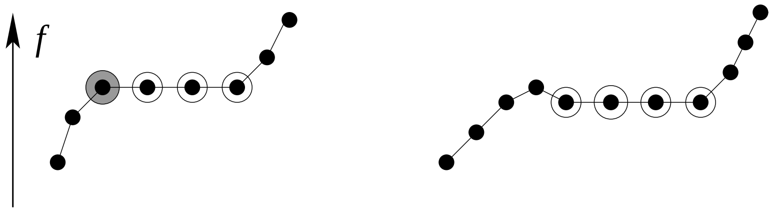


Figure 19: Two types of local minima: shoulder (l.h.s) and valley (r.h.s.). Local minima are marked by white circles. The vertex marked in gray is a saddle point but not a minimum of the landscape.

Definition 2. *Consider a landscape* f *on* $G(X, E)$*. We say* (G, f) *is*

non-degenerate *or* invertible *if* $f(x) = f(y)$ *implies* $x = y$ *for all* $x, y \in X$*;*
locally invertible *if* $x, y \in \mathcal{N}(z)$ *and* $f(x) = f(y)$ *implies* $x = y$ *for all* $z \in X$*;*
non-neutral *if* $f(x) = f(y)$ *and* $y \in \mathcal{N}(x)$ *implies* $x = y$ *for all* $y \in X$*;*

We collect a number of obvious consequences of definition 2 in the following

Lemma 3. *(1) non-degenerate* $\Longrightarrow$ *locally invertible* $\Longrightarrow$ *non-neutral*
(2) If (G, f) *is locally invertible then the end-point of a gradient walk is uniquely determined by its initial condition. (3) All local minima are strict in non-neutral landscapes.*

Walks A *walk* $\mathbf{p}$ of length m on a graph G is a sequence

$$[x_1, e_1, x_2, e_2, \ldots, x_m, e_m, x_{m+1}] \text{ with } x_i \in X,\ e_i \in E,\ \text{ and } e_i = \{x_i, x_{i+1}\}.$$

A walk is called a path if all x_i and all e_i are distinct. Intuitively, a saddle point on the way from x to y is a maximum along a walk from x to y that is as low as possible. Although one usually works with paths in a graph-theoretical setting, we will use walks in the present context

Definition 4. *Let $\mathbb{P}_{xy}$ be the set of all walks from x to y. We say that x and y are* mutually accessible at level η, *in symbols*

$$x \overset{\eta}{\longleftrightarrow} y\,, \tag{15}$$

if there is walk $\mathbf{p} \in \mathbb{P}_{xy}$ such that $f(z) \leq \eta$ for all $z \in \mathbf{p}$, respectively.
The relation $\overset{\eta}{\longleftrightarrow}$ is obviously symmetric ($x \overset{\eta}{\longleftrightarrow} y$ implies $y \overset{\eta}{\longleftrightarrow} x$) and transitive ($x \overset{\eta}{\longleftrightarrow} y$ and $y \overset{\eta}{\longleftrightarrow} z$ implies $x \overset{\eta}{\longleftrightarrow} z$). It is reflexive for all $\eta \geq f(x)$. The following property will be used repeatedly:

Lemma 5. *For all $x, y, z \in X$:*

$$x \overset{f(x)}{\longleftrightarrow} y \quad \textit{and} \quad y \overset{f(y)}{\longleftrightarrow} z \quad \textit{implies} \quad x \overset{f(x)}{\longleftrightarrow} z \tag{16}$$

Proof. Observe that $x \overset{f(x)}{\longleftrightarrow} y$ implies $f(y) \leq f(x)$; hence $y \overset{f(x)}{\longleftrightarrow} z$. □

Definition 6. *The* saddle height $\hat{f}(x, y)$ *between two configurations $x, y \in X$ is the minimum height at which they are accessible from each other, i.e.,*

$$\hat{f}(x, y) = \min_{\mathbf{p} \in \mathbb{P}_{xy}} \max_{z \in \mathbf{p}} f(z) = \min\{\eta | x \overset{\eta}{\longleftrightarrow} y\} \tag{17}$$

In particular, we have $\hat{f}(x, x) = f(x)$.

Cycles The notion of cycles from the theory of simulated annealing [21, 22] reduces to connected components of level sets for our purposes. The valleys $\mathcal{V}(s)$ discussed in section 3.2 are special cases of cycles.

Definition 7. *The* cycle of $x \in X$ at height η, $C_\eta(x)$, *is the connected component of the level set $\{y \in X | f(y) \leq \eta\}$ that contains x.*

Thus $x \in C_\eta(x)$ for $\eta \geq f(x)$ and $C_\eta(x) = \emptyset$ for $\eta < f(x)$. Obviously, we have

$$C_\eta(x) = \left\{ y \in X \middle| y \overset{\eta}{\longleftrightarrow} x \right\} \tag{18}$$

Lemma 8. *Let $\eta' \leq \eta''$ and $x, y \in X$. Then either $C_{\eta'}(x) \cap C_{\eta''}(y) = \emptyset$ or $C_{\eta'}(x) \subseteq C_{\eta''}(y)$.*

Proof. Suppose there is $q \in C_{\eta'}(x) \cap C_{\eta''}(y)$. Then $x \overset{\eta'}{\longleftrightarrow} q \overset{\eta''}{\longleftrightarrow} y$, hence $x \overset{\eta''}{\longleftrightarrow} y$ and $C_{\eta'}(x) \subseteq C_{\eta''}(x)$ because of lemma 5. $\square$

As an immediate consequence we note:

Corollary 9. *The set $\mathcal{C}(G, f) = \{C_\eta(x) | x \in X, \eta \in \mathbb{R}\}$ of the cycles of the landscape (G, f) forms a* hierarchy, *i.e., for all $x, y \in X$ and $\eta', \eta'' \in \mathbb{R}$ we have either $C_{\eta'}(x) \cap C_{\eta''}(y) = \emptyset$, $C_{\eta'}(x) \subset C_{\eta''}(y)$, $C_{\eta''}(y) \subset C_{\eta'}(x)$, or $C_{\eta'}(x) = C_{\eta''}(y)$.*

Commute Points Next we identify a set of points that is closely related to our intuition of a "saddle point".

Definition 10. *The point $z \in X$ is a* commute point *between $x \in X$ and $y \in X$ if there is a walk $\mathbf{p} \in \mathbb{P}_{xy}$ such that*

(o) $z \in \mathbf{p}$;
(i) $\max_{u \in \mathbf{p}} f(u) = \hat{f}(x, y) = f(z)$.

The set of commute points between x and y is denoted by $S^*(x, y)$.

Commute points can also be characterized in terms of the cycles introduced above.

Theorem 11. *A point $s \in X$ is a commute point between x and y if and only if*

(i) $C_\eta(x) \cap C_\eta(y) = \emptyset$ *for all* $\eta < f(s)$.

(ii) $C_{f(s)}(x) = C_{f(s)}(y)$

(iii) $s \in C_{f(s)}(x)$

Proof. First suppose s is a commute point between $x, y \in \mathcal{M}$. Thus $x \overset{f(x)}{\longleftrightarrow} y$ and hence $C_{f(s)}(x) = C_{f(s)}(y)$. Since $f(s) = \max_{u \in \mathbf{P}} f(u) = \hat{f}(x, y)$ we know that $x \overset{\eta}{\not\longleftrightarrow} y$ for all $\eta < f(s)$, i.e., (i) holds. Condition (o) of the definition of course implies (iii) of the theorem.
Conversely, assume the conditions of the theorem. By (ii) we have $x \overset{f(s)}{\longleftrightarrow} y$ and hence there is $\mathbf{p} \in \mathbb{P}_{xy}$ with height $\eta \geq f(s)$. From (i) we see that there is no such path with height $\eta < f(x)$, thus $\hat{f}(x, y) = \max_{u \in \mathbf{p}} f(u) = f(s)$. Since $s \in C_{f(s)}(x)$ we can choose $\mathbf{p}$ to run through s. $\square$

The definition of $S^*(x,y)$ might seem strange since it defines the end-points x and y of the walk as commute points whenever they are the highest points along some $\mathbf{p} \in \mathbb{P}_{xy}$. In particular we have $x \in S^*(x,x)$. These properties are, however, a significant technical convenience.

Strict Merging Points While definition 10 above is appealing because of its (relative) simplicity, it has a major shortcoming in the degenerate case. Consider the simple 1-dimensional landscape on the l.h.s. of Figure 20. Then both a and b are commute points according to definition 20, a fact that contradicts the intuitive notion of saddle points.A different approach starts from the cycles instead of considering walks connecting local minima.

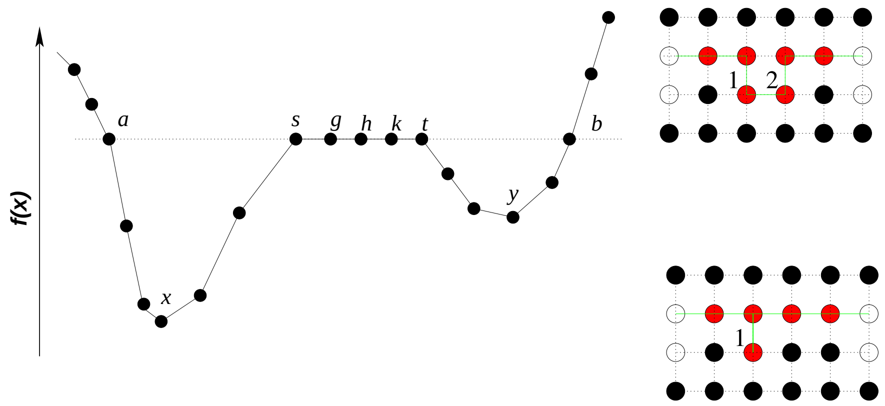


Figure 20: Saddle points in a degenerate landscape. L.h.s.: All of a, b, g, h, k, s, t are commute points, g, h, k are (degenerate) local minima, g, h, k, s and t are weak merging points (and hence also saddle points), none of these points is a (strict) merging point.
R.h.s.: f-values increase from white to black. The red vertices are saddle points connecting the two white basins. Replacing walks by paths in definition 15 would imply that 1 is not saddle point between x and y in the second situation, while it is such a saddle in the first situation. While this might be geometrically appealing it causes algorithmic difficulties since we cannot treat the flat connected part as a unit.

Definition 12. *A point $m \in X$ is a* strict merging point *if there are local minima $x, y \in \mathcal{M}$ such that*

(i) $C_\eta(x) \cap C_\eta(y) = \emptyset$ *for all* $\eta < f(m)$,

(ii) $C_{f(m)}(x) = C_{f(m)}(y)$,

(iii) $m \in \overline{C_{\eta'}(x)} \cap \overline{C_{\eta''}(x)}$ *for some* $\eta', \eta'' < f(m)$.

We say that m is a strict merging point between (the basins of) x and y in this case. The set of strict merging points between x and y will be denoted by $\hat{M}(x,y)$.

If f is non-degenerate, then m is a strict merging point between x and $y \neq x$ if and only if it is the (uniquely determined) commute point between x and y.

Lemma 13. *If m is a strict merging point then it is a commute point. If m is a strict merging point, then it is not a local minimum.*

Proof. From (i) and (ii) we conclude that $f(m) = \hat{f}(x,y)$. By (iii) m is in particular at least a neighbor of a point $q \in C_{f(m)}(x)$ and thus itself contained in the connected component $C_{f(m)}(x)$. Hence condition (iii) of definition 11 is satisfied.
To see the second part of the lemma consider local minimum z. Then $C_\eta(z) = \emptyset$ for all $\eta < f(z)$ and hence $\overline{C_h(z)} = \emptyset$, and condition (iii) is never satisfied. □

In a non-degenerate landscape each commute point is therefore either a local minimum or a strict merging point.

Merging Point None of the points s, t, g, h, k in the l.h.s. of Fig. 20 is a strict merging point. Thus, strict merging points are also not the desired construction. By lemma 13 we see that the desired definition must lie somewhere between commute points and strict merging points. Let us consider both avenues.

In order to weaken the definition of strict merging points we define the *borderless cycle of x at height η* in the following way

$$C^\circ_\eta(x) = \begin{cases} \bigcup_{\eta' < \eta} C_{\eta'}(x) & \text{if } \eta > f(x) \\ \{x\} & \text{if } \eta = f(x) \quad \text{and } x \in \mathcal{M} \\ \emptyset & \text{otherwise} \end{cases} \tag{19}$$

Definition 14. *$m \in X$ is a* merging point *between the local minima $x, y \in \mathcal{M}$ if*

(i) $C_\eta(x) \cap C_\eta(y) = \emptyset$ *for all* $\eta < f(m)$,

(ii) $C_{f(m)}(x) = C_{f(m)}(y)$

(iii) $m \in \overline{C^\circ_{f(m)}(x)} \cap \overline{C^\circ_{f(m)}(y)}$

We write $M(x,y)$ for the set of merging points between x and y. The main difference between definitions 14 and 12 is that $z \in \mathcal{M}$ is always a merging point but never a strict merging point. To see this, choose $x = y = z$. Then (i) and (ii) is satisfied trivially, and (iii) follows from $C^{\circ}_{f(z)}(z) = \{z\}$. Furthermore, the definition now includes s and t in Fig. 20 as merging points between x and g, and k and y, respectively.

Saddle Points The definition of a commute point includes a and b in Fig. 20 because nothing prevents the walk connecting x with y from first visiting a and returning to x before crossing the "true saddle" to y. With the help of the the following notation we can "repair" this definition. The idea is now to consider only walks that never return to a basin that they have already left. The following formalization(s) of this idea appears natural:

Definition 15. *A point s is* saddle point *between x and y if there is a walk $\mathbf{p} \in \mathbb{P}_{xy}$ such that*

(o) $s \in \mathbf{p}$;

(i) $\max_{u\in\mathbf{p}} f(u) = \hat{f}(x,y) = f(s)$;

(ii) For all $z \in \mathcal{M}$: $G[C^{\circ}_{f(s)}(z) \cap \mathbf{p}]$ is connected.

A saddle point between x and y is direct *if the walk $\mathbf{p}$ in addition satisfies*

(iii) $G[\{u \in \mathbf{p} | f(u) = f(s)\}]$ *is connected.*

We write $S(x,y)$ and $\hat{S}(x,y)$ for the saddle points and direct saddle points between x and y, respectively.

Condition (ii) ensures that $\mathbf{p}$ meets the inside of a basin not more than once. Condition (iii) means that the walk can be chosen as "unimodal", leading from $C^{\circ}_{f(s)}(x)$ to $C^{\circ}_{f(s)}(y)$ in such a way that $f(u) = f(s)$ for all configurations in between. The following simple result *a posteriori* justifies the name "saddle height" for $\hat{f}(x,y)$ introduced in definition 6.

Lemma 16. *For all $x, y \in \mathcal{M}$ there is a saddle point $s \in S(x,y)$ such that $f(s) = \hat{f}(x,y)$.*

Proof. The basins $C^{\circ}_{f(s)}(z)$ are connected by construction. Thus, a walk $\mathbf{p}$ satisfying (ii) can be obtained from any walk $\mathbf{p}'$ connecting x and y at level $f(s)$ by replacing the part beginning at the first point $z' \in \mathbf{p}' \cap C^{\circ}_{f(s)}(z)$ to the last point $z'' \in \mathbf{p}' \cap C^{\circ}_{f(s)}(z)$ along $\mathbf{p}'$ by a walk from z' to z'' that is contained entirely in $C^{\circ}_{f(s)}(z)$. The restriction of definition 6 to walks satisfying condition (ii) hence does not affect the saddle point height $\hat{f}(x, y)$. □

Obviously, every saddle point is a direct saddle point between *some* basins, but not all pairs of minima are connected by a direct saddle.

Theorem 17. *If $m \in X$ is a merging point then it is a saddle point. If $x \neq y$ then $M(x, y) \subseteq \hat{S}(x, y)$ for all $x, y \in \mathcal{M}$.*

A proof for that can be found in [49].

The following relationships for all $x, y \in \mathcal{M}$ follow directly from the definitions:

$$\hat{M}(x, y) \subseteq M(x, y)$$
$$\hat{S}(x, y) \subseteq S(x, y) \subseteq S^*(x, y)$$

For $x \neq y$ we have $M(x, y) \subseteq \hat{S}(x, y)$ while $M(x, x) = \{y \in \mathcal{N}(x) | f(x) = f(y)\}$ and $\hat{S}(x, x) = \{x\}$.

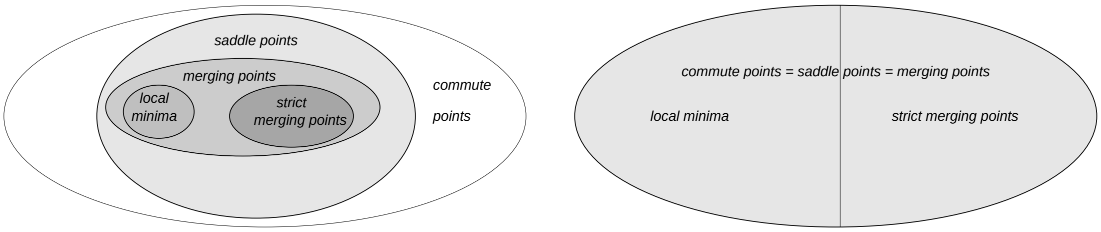


Figure 21: Venn-diagrams of the various notions of saddle points; l.h.s: degenerate landscapes, r.h.s: non-degenerate landscapes.

Naturally, we define the set of all Q-points of a landscape, where Q is one of $S^*, S, \hat{S}, M, \hat{M}$, as

$$Q(G, f) = \bigcup_{x,y \in \mathcal{M}} Q(x, y) \tag{20}$$

i.e. S is a Q-point if and only if it is a Q-point between two local minima. The mutual relationships between these sets and the set $\mathcal{M}$ of local minima are summarized in Fig. 21.

Equivalent Saddle Points Just as in the case of local minima it is useful to collect Q-points into equivalence classes. There appear to be two natural equivalence relations

Definition 18. *Two Q-points $x, y \in Q$ are*

- $\leftrightarrow$ weakly equivalent *if $f(x) = f(y)$ and $x \overset{f(x)=f(y)}{\longleftrightarrow} y$.*
- $\Leftrightarrow$ equivalent *if they lie in the same connected component of the weak equivalence classes.*

We write Q^* and $Q(x)$ for the weak equivalence class and equivalence class of Q-points that contains x belongs.

Note that this definition also applies to the local minima.

Lemma 19. *(i) For all $x \in \mathcal{M}$ we have $\mathcal{M}(x) \subseteq S(x)$ and $\mathcal{M}^*(x) \subseteq S^*(x)$.*
(ii) $\mathcal{M}(x) = \mathcal{M}^*(x) = S(x) = S^*(x)$ if and only if $\mathcal{M}(x)$ is a valley.

Proof. (i) follows immediately from the fact that every local minimum is also a saddle point.
(ii) Suppose $\mathcal{M}(x)$ is a valley, i.e., $f(y) > f(x)$ for all $y \in \partial\mathcal{M}(x)$. Consider a vertex $z \in X \setminus \mathcal{M}(x)$. Each walk $\mathbf{p}$ connecting $u \in \mathcal{M}(x)$ with z must pass through a vertex $q \in \partial\mathcal{M}(x)$, hence $\hat{f}(u, z) \geq \min_{q \in \partial\mathcal{M}(x)} f(q) > f(x)$, i.e., $z \not\overset{f(x)}{\longleftrightarrow} u$ and hence $z \notin S^*(x)$. Thus $S^*(x) \subseteq \mathcal{M}(x)$.
The converse follows from (i). $\square$

In the remainder of this section we will show that barrier trees can be formulated in terms of weak equivalence classes.

Formal definition of barrier trees

Let us write $\mathfrak{S} = \{S^*(x) | x \in S\}$ and $\mathfrak{M} = \{\mathcal{M}^*(x) | x \in \mathcal{M}\}$ for the sets of weak equivalence classes of saddle points and local minima respectively. It is the purpose of this section to show that the set $\mathfrak{U} = \mathfrak{M} \cup \mathfrak{S}$ of equivalence classes of saddles and minima can be regarded in a very natural way as the vertex set of a tree.

We observe that the cost function f is by construction constant on each set $U \in \mathfrak{U}$; hence we write $f(U)$ instead of "$f(x)$ for each $x \in U$". Thus the notation $U' \overset{\eta}{\longleftrightarrow} U''$ is also well defined. In the same vein we may write $\hat{f}(W, W')$ instead of "$\hat{f}(w, w')$ for each pair $(w, w') \in W \times W'$.

We will need the following subsets of $\mathfrak{U}$ in our discussion:

$$\mathfrak{U}(W) = \begin{cases} \{W' \in \mathfrak{U} | W' \overset{f(W)}{\longleftrightarrow} W\} & \text{for} \quad W \in \mathfrak{S} \setminus \mathfrak{M} \\ \{W\} & \text{for} \quad W \in \mathfrak{M} \end{cases} \tag{21}$$

The properties of this collections of sets is summarized in the following

Lemma 20. *(i) For all $W' \in \mathfrak{U}(W)$ we have either (a) $f(W') < f(W)$, or (b) $W = W'$, or (c) $W' \in \mathfrak{M}$ and $W' \subset W$.*

(ii) $\{\mathfrak{U}(W) | W \in \mathcal{U}\}$ is a hierarchy.

(iii) For all $W', W'' \in \mathfrak{U}$ there is a unique $W \in \mathfrak{U}$ such that (1) $W', W'' \in \mathfrak{U}(S)$ and (2) if $W''' \in \mathfrak{U}(W)$ and $W', W'' \in \mathfrak{U}(W''')$ then $W = W'''$. We have $\hat{f}(W', W'') = f(W)$.

Proof. (i) By construction W' is accessible from W at level $f(W)$. Thus $f(W') \leq f(W)$. If $f(W') < f(W)$ then $W \cap W' = \emptyset$, i.e., none of (b) or (c) can holds. Now suppose $f(W) = f(W')$. If $W \in \mathfrak{M}$ then by construction $\mathfrak{U}(W) = \{W\}$, i.e., (b) holds. If $W \in \mathfrak{S} \setminus \mathfrak{M}$ then W is the saddle point set of a shoulder, and hence it contains the minima set W' of the shoulder by lemma 19. Clearly $W \overset{f(W)}{\longleftrightarrow} W'$, hence alternative (c) is satisfied. Following the argument in lemma 9 one easily verifies that $\{\mathfrak{U}(W) | W \in \mathcal{U}\}$ is a hierarchy. Hence for each $W', W'' \in \mathfrak{U}$ there is an $W \in \mathfrak{U}$ such that $W', W'' \in \mathfrak{U}(W)$ because the graph G is connected. The existence and uniqueness of the minimal element W is now obvious. Clearly W is the set of saddle points connecting W' and W'', thus $f(W)$ is the height of these saddle points. $\square$

Recall that, given a collection $\mathcal{A}$ of sets, $A \in \mathcal{A}$ is a maximal subset of B if $A \subseteq B$ and there is no $A' \in \mathcal{A}$ such that $A \subset A' \subseteq B$.

The *children* of $W \in \mathfrak{U}$ form the set

$$\mathsf{children}(W) = \big\{W' \in \mathfrak{U}(W) \big| \mathfrak{U}(W') \text{ is a maximal subset of } \mathfrak{U}(W) \setminus \{W\}\big\} \tag{22}$$

Let $\mathfrak{T}(G, f)$ be the graph with vertex set $\mathfrak{U}$ and a directed edge $(W', W'') \in \mathfrak{E}$ if and only if $W'' \in \mathsf{children}(W')$.

Theorem 21. *The graph $\mathfrak{T}(G,f)$ is a rooted tree.*

Proof. It follows directly from eq.(21) that $W \in \mathfrak{U}$ has a non-empty set of children if and only if $W \notin \mathfrak{M}$. Furthermore, lemma 20 implies that each W' is the child of at most one $W \in \mathfrak{U}$. Lemma 20 also implies that either $f(W') < f(W)$ or $W' \in \mathfrak{M}$, in which case W' has no children. Thus $\mathfrak{T}(G,f)$ is acyclic. The hierarchy property ensures that $\mathfrak{T}(G,f)$ is a rooted forest. Finally, since X is finite and $G(X,E)$ is connected we have $x \overset{\max f}{\hookleftarrow\!\!\!\hookrightarrow} y$, i.e., there is $W^* \in \mathfrak{U}$ such that $\mathfrak{U}(W^*) = \mathfrak{U}$. Thus $\mathfrak{T}(G,f)$ is a connected rooted forest, i.e., a rooted tree. □

We call $\mathfrak{T}(G,f)$ the *barrier tree* of the landscape (G,f). If f is non-degenerate, then each vertex of $\mathfrak{T}(G,f)$ is a set consisting of a single local minimum or saddle point. With each edge (W,W') of $\mathfrak{T}(G,f)$ we associate the difference in the cost function $f(W) - f(W')$.

Merging Graph Closely related to the barrier tree is the *merging graph* of a landscape (figure 22). Its vertices are the local minima and the equivalence classes $Q(x)$ of saddle points. A local minimum y is connected to $Q(x)$ if there is $z \in \mathcal{M}$ and $u \in Q(x)$ such that u is a merging point between z and y. The merging graph is usually not a tree. For instance, there may be more than one connected components of saddle points that merge the same two minima. Hence merging graphs have in general more nodes than the corresponding barrier trees.

The calculation of degenerate landscapes (algorithm 2) is implemented similar to the non-degenerate version (algorithm 1) with the difference that structures are processed as sets of equal-energy structures (energy bands): A global list $\mathcal{B}$ contains *all* local minima found during the procedure. $\mathcal{C}$ is the list of connected components of the current energy-band[11]. Both $\mathcal{B}$ and $\mathcal{C}$ must be empty at the beginning of the algorithm. The calculation starts from an energy-sorted list of configurations that are processed in terms of energy bands (line 2). We will further require a *local* list of minima, $\mathcal{K}$ as well as a temporary set $\mathcal{C}_{temp}$. For each element of an energy band, d, all neighbor structures according to a predefined

[11] with respect to the move-set

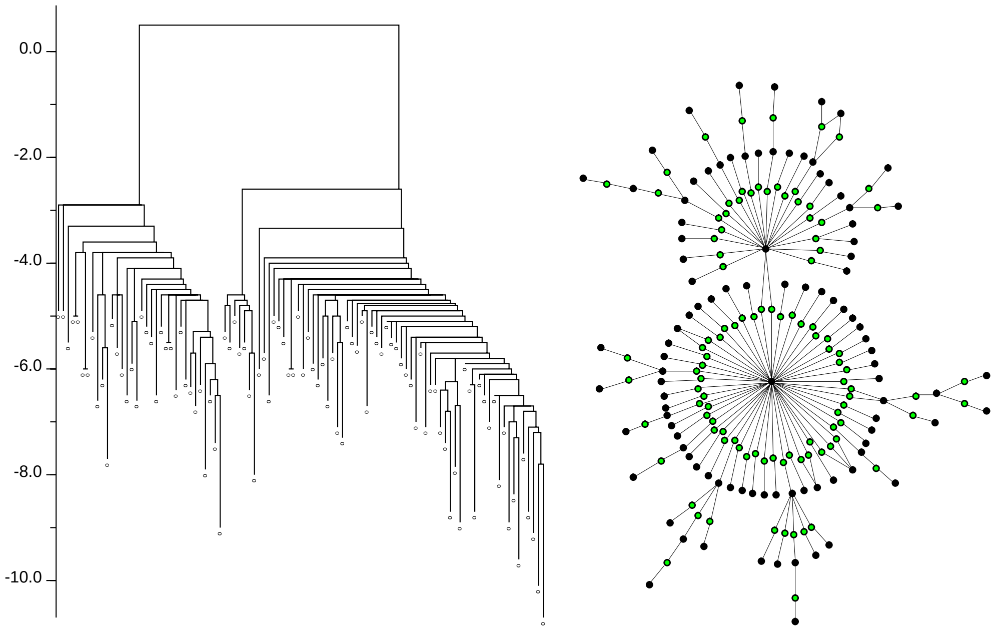


Figure 22: Energy Landscape of SL RNA of *Leptomonas collosoma*. We show the barrier trees (l.h.s.) and the merging graphs (r.h.s.) for comparison. The example is restricted to 100 lowest local minima.

move-set are calculated and stored on the neighbor-stack (line 7). The neighbor-stack is then processed and a routine searches a hash if any neighbor of d has been seen in a previous step of the computation. If this is true, the according basins are stored in $\mathcal{K}$. If y has the same energy as d, d is added to the component containing y and $\mathsf{c}(y)$ is added to $\mathcal{C}_{temp}$ (lines 12 and 13). There are two possibilities after all neighbors y have been processed: Either $\mathcal{C}_{temp}$ is empty, which means that a new component containing d is opened (line 18). In case $\mathcal{C}_{temp}$ contains elements, they are merged with $\mathcal{C}$ (line 20) and the local list of basins that are connected by $\mathsf{c}(d)$ is updated (line 21). After all elements d of the current energy-band have been processed, $\mathcal{K}$ is checked if it is empty (indicating that d is a new local minimum) and, if this is true, the global list of local minima is updated (line 25). Finally, d is written to the hash (line 27). The last step of the algorithm is achieved by merging basins connected by individual components $\mathsf{c} \in \mathcal{C}$.

Basins Let $B(S) = \{x \in X | x \overset{f(S)}{\longleftrightarrow} S \text{ and } f(x) < f(S)\}$, and let $B(S';S)$ be the vertex set of the connected component of $G[B(S)]$ that contains S'. Thus we have $B(S';S) = \emptyset$ if and only if $S' \notin \mathfrak{U}(S) \setminus S$. The set $B(S';S)$ is the *basin* of

Algorithm 2 Variant of barriers that generates saddle point components

Require: subopt, $\mathcal{B}, \mathcal{C}$

$\mathcal{B} \Leftarrow \emptyset$
while $\mathcal{D} \Leftarrow \mathit{read\_energy\_band}()$ **do**
    for all $d \in \mathcal{D}$ **do**
        $\mathcal{K} \Leftarrow \emptyset$
        $\mathcal{C} \Leftarrow \emptyset$
        $\mathcal{C}_{temp} \Leftarrow \emptyset$
        $\mathcal{N} \Leftarrow \mathit{generate\_neighbors}(d)$
        for all $y \in \mathcal{N}$ **do**
            if $b \Leftarrow \mathit{lookup\_hash}(y)$ **then**
                $\mathcal{K} \Leftarrow \mathcal{K} \cup b$
                if $\mathrm{E}(y) = \mathrm{E}(d)$ **then**
                    $\mathsf{c}(y) \Leftarrow \mathsf{c}(y) \cup \{d\}$
                    $\mathcal{C}_{temp} \Leftarrow \mathcal{C}_{temp} \cup \{\mathsf{c}(y)\}$
                end if
            end if
        end for
        if $\mathcal{C}_{temp} = \emptyset$ **then**
            $\mathcal{C}_{temp} \Leftarrow \{\{d\}\}$
        else
            $\mathcal{C} \Leftarrow \mathit{merge\_components}(\mathcal{C}, \mathcal{C}_{temp})$
            $\mathcal{K}(\mathsf{c}(d)) \Leftarrow \mathcal{K}(\mathsf{c}(d)) \cup \mathcal{K}$
        end if
    end for
    if $\mathcal{K} = \emptyset$ **then**
        $\mathcal{B} \Leftarrow \mathcal{B} \cup \{d\}$
    end if
    $\mathit{write\_hash}(d)$
end while
for all $\mathsf{c} \in \mathcal{C}$ **do**
    $\mathit{merge\_basins}(\mathcal{K}(\mathsf{c}))$
end for

S' in the subtree with root S. We have $B(S';S) = B(S'';S)$ if and only if there is $S''' \in \mathfrak{U}(S) \setminus \{S\}$ such that $S', S'' \in \mathfrak{U}(S''')$, i.e., if S' and S'' are connected by a saddle S''' within the basin below S. The basins at or below a node S of the tree of course form a partition of the corresponding level sets and hence can be used to define macro-states. Note that this definition is highly different from the "valley $\mathcal{V}$" in section 3.2. The problem with this approach is, however, that the partition depends explicitly on the energy level; there does not appear to be a natural way to extend such a partition of a level set to a partition of the complete state space.

A second type of basin is defined by gradient walks. In locally invertible landscapes gradient walks are uniquely defined. Hence, for each starting point $x \in X$ there is a uniquely defined end point $g(x) \in \mathcal{M}$. Naturally, we define for each $z \in \mathcal{M}$ the associated gradient basin as $\mathcal{B}(z) = \{x | z = g(x)\}$. Obviously $x \in \mathcal{B}(x)$. Furthermore, the gradient basins form a natural partition of the complete state space X. This partition is consistent with the barrier tree in the following sense:

$$\mathcal{B}(x) \cap \{y | f(y) < f(S)\} \subseteq B(S';S) \text{ for } x \in S' \tag{23}$$

That is, the part of the gradient basin of x below the level $f(S)$ is contained in the basin $B(S';S)$ of the minimum x within the subtree with root S.

5 The protein folding problem

At this point we have established a framework that allows us to investigate the energy landscape of biopolymers on a theoretical level. We have introduced the fundamental concept of energy landscapes and presented tools for efficient calculation of the features of energy landscapes, such as number of local minima, sizes of basins of attraction and energy barriers separating those basins. A major motivation for this thesis was the ability to investigate energy landscapes and dynamics of proteins. More exactly, we wanted to address the question *what* makes a protein fold to its native state and what are the forces that drive protein folding? We will give an overview of past and present understanding of protein folding in the following (see [34, 35, 136] for more details).

The tertiary structure of a protein is crucially determined by its amino acid sequence. This was demonstrated by denaturation experiments showing that denaturation of some proteins is reversible. Certain globular proteins that have been denatured by heat, denaturing reagents or extremes of pH regain their native structure and biological activity if they are returned to conditions in which the native conformation is stable. This process, called renaturation, was first shown by Anfinsen in the 1950ies. It provided first evidence that the amino acid sequence of a polypeptide chain contains all information necessary to fold the chain to its native, three dimensional structure.

When Pauling an Mirsky proposed that backbone hydrogen bonding is a prominent driving force [112], a *backbone-centric, helix-centric* view of protein folding was born. This view should be the major viewpoint for protein folding for almost half a century, from the 1930ies to the 1980ies. In the same time period, folding cooperativity was elucidated by an understanding of helix-coil transitions, see e.g [134] . With cooperativity we mean that there is a dramatic transition from denatured to native states upon only small changes in pH, solvent or temperature. It became clear that helix-coil transition is driven by hydrogen bonding and $\phi\psi$ propensities among near-neighbor groups along the chain. Another prominent aspect arose within this view, namely that protein folding should be hierarchical: the primary sequence leads to secondary structure (fast), which is then assembled into tertiary structure (slower). Hydrogen bonding and $\phi\psi$ propensities were seen as a large part of the explanation of the structures. Interestingly, hydrophobicity was seen as some 'nonspecific' force that aided a polymer to collapse but otherwise had little interference in guiding a protein to its native state. Rather, hydrogen bonding and helical propensities were seen as the major driving force.

Within the last 20 years, a different view arose, a *side-chain-centric* one. In this view, the greater contribution to the free energy of folding is encoded in a more delocalized 'solvation' code rather than in propensities for nearest neighbor amino acids to favor certain $\phi\psi$ values. The main idea behind this view is the fact that only a small fraction of all possible conformations can bury nonpolar residues to the greatest possible degree. It became evident that hydrophobic interactions are among the strongest interactions among amino acids in water. Hence, hydrophobic forces are no longer seen as nonspecific 'glue' as in the backbone-centric view, but as a crucial, structure-determining force. In this view, folding cooperativity more closely resembles a process of polymer collapse than a helix-coil transformation [34].

It still remains so specify which view is the more 'biological' one as the true balance between side-chain and backbone forces is not yet known. One one hand, a simplified side-chain centric model can predict properties of globular proteins, on the other hand not all proteins do collapse and it is still a $\phi\psi$-based model that explains this kind of behavior. Nevertheless, the side-chain-centric view tends to be the more general one since helix and strand propensities are believed to be rather weak. Further, experiments showed that protein folding is not hierarchical, which means that secondary structure is not necessarily a building block for tertiary assembly [67]

Much knowledge on the theoretical background of protein folding and dynamics has been achieved within the last 30 years by a rigorous assessment of simplified models (section 2.5). We will make use of the HP model for the protein case throughout this thesis. For small sequences, reduced models like this allow not only for an extensive exploration of the conformation space, but also provide a reasonable means to calculate e.g. thermodynamic properties that could not be calculated in any other way, but that can be tested by experiments.

Two key problems have often been reported in literature and should thus be mentioned here[12]:

[12] though they are both titled 'Paradox', polymer modeling demonstrated that they are neither paradox nor problematic.

- **The Blind Watchmaker Paradox**: The probability to find natural proteins by a random search in sequence space is vanishingly small.
- **The Levinthal Paradox**: The probability that a protein is able to find its unique, native state by a random search through conformation space is impossibly small.

In both cases, the key to impossibility is given by the vastness of the underlying search space. Consider a polypeptide chain of length 100. Assuming there are 20 different amino acids, there are some $20^{100} = 10^{130}$ different sequences of this length. Evidently it is impossible that nature could have searched through such a sequence space to find a certain sequence. In fact, what *is* relevant for a biologically relevant protein is its fold, *not* its sequence. It turned out that the probability to find *any* sequence that folds to a specific structure from a large ensemble soup (still considering our 100 residue-polypeptide) is nearly 100 orders of magnitude larger (roughly 10^{-10} to 10^{-20}) than finding a very specific sequence (probability of about 10^{-130}). This is due to an enormous 'degeneracy' in sequence space: many different sequences can fold to the same (native) structure. Within the context of the simple HP model (assuming that the 3D structure is encoded in the binary sequence) we can reduce our sequence space from 20^{100} to $2^{100} = 10^{30}$. Though these are still too may sequences to investigate exhaustively within the currently available computational framework, a simple consideration suggests that hydrophobic monomers are largely interchangeable within each other, as well as polar monomers are. Moreover, studies in the 1990ies [25] showed that only a fraction (1/3 referring to [34]) of monomers are essential for folding, those that define a hydrophobic core of the model protein. So if one is only interested in native model protein structures, the sequence space is again decreased from 2^N to $2^{N/3}$ ($2^{33} = 10^{10}$ for $N = 100$). The general view for the ensemble of sequence space is that virtually all molecules are "nearly" folded, i.e. a random chain of length 100 is assumed to be highly compact in water, have considerable secondary structure elements and is structured much like a "molten globule".

At this point is is necessary to state that native protein structures are not perfect spheres. Globular proteins have hydrophobic cores and thus are highly, but not maximally compact. In fact, the deviations from perfect compactness in global shape, active sites and surface cavities are intrinsic to protein structure and function. The HP model accounts for this as native states within this model are often not maximally compact. The shape of any state in the HP model is dependent on its monomer sequence.

In the remainder of this section we will focus on some aspects of protein folding that first appeared in literature in the late 1980ies and early 1990ies: The concept of *folding funnels* [16, 36, 92] which has often been used for a qualitative description of the protein folding process. Although the idea behind the concept of folding funnels is appealing, a rigorous mathematical formulation for the model is, to our knowledge, still missing.

When Levinthal proposed his thoughts in the late 1960ies [93], the general opinion was that two - mutually exclusive - options should play an important role in protein folding: On the one side *thermodynamic control*, indicating that a protein should reach its global minimum energy via a pathway-independent folding mechanism (assuming the native structure is determined only by final native conditions). On the other side, *kinetic control* which accounts for quick folding due to pathway-dependence. Under kinetic control, intermediate states (whether they are on- or off-pathway) should be responsible for a rapid formation of the native state. Within the following years, the door was opened for many kinetic experiments that were seen as the key to elucidating the "folding code".

Experimentalists often formulate their findings in protein kinetics in terms of mass-action diagrams with arrows that connect certain states with each other: D (denatured), I (intermediate) and N (native). These states do *not* correspond to single structures of a (model) protein, but an ensemble (or macrostate). Such schemes describe observed relaxation rates and amplitudes and are sometimes denoted the "old" or "classical" view in protein folding. The classical view (also known as "sequential micropath view") is based on simple phenomenological kinetic models that are derived from single- or multiple-exponential time decays of optical properties that monitor changes in the protein structure after a jump to folding or unfolding conditions [36].

However, scientists were not quite satisfied with this level of description as it is not capable of describing the process of protein folding on a microscopic level. It rather explains the 'average' behavior of a protein. Further, this model suggests that all polypeptide chains must follow the same pathway to find its native state (which is obviously not the case).

The solution to the problem was seen in introducing a novel view of protein folding, namely an "ensemble" view. Instead of trying to address protein folding in terms of macroscopic states (see above), a different approach relying on the whole ensemble of possible structures (based on statistical mechanic modeling) was proposed. Within this view, literature argues (see e.g. [35]) that more attention is put into the question 'what molecules do' - instead of 'what exponentials do'. The ensemble view suggests that pathways of sequential events should be replaced with a funnel concept of parallel events: Polymer chains are supposed to fall energetically downhill, as when balls roll down bumpy funnels. Hydrophobic collapse leads to many different compact chain conformations. According to [34], the folding funnel arises because the 'drive to collapse is also a drive toward a reduced ensemble of conformation'. In other words, there are many non-native, high-energy states and only one (or at least not more than a couple of) native states with very low energy.

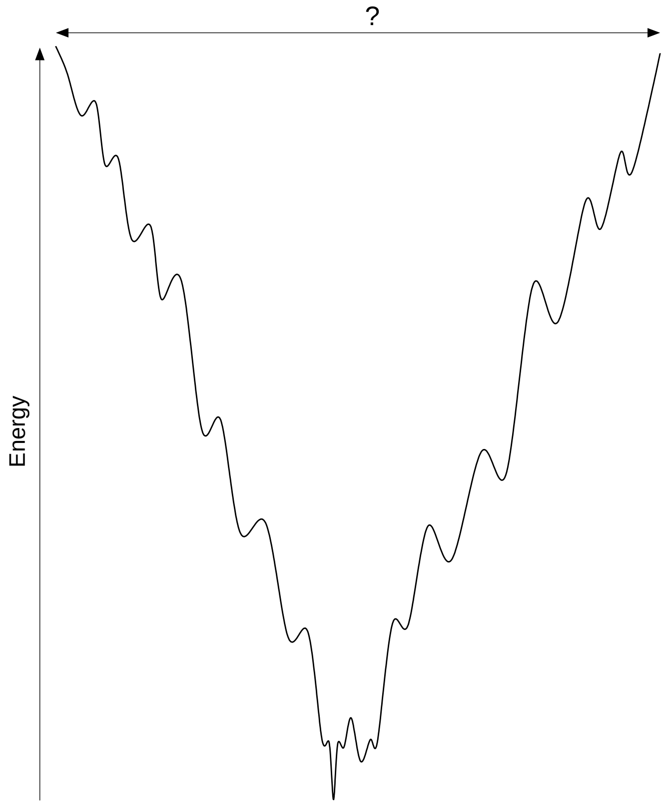


Figure 23: An idealized folding funnel.

Folding funnels are often illustrated as in figure 23. The vertical axis is said to represent the energy of a given chain configuration: Torsion angle energies, ion pairs, hydrogen bonds, hydrophobic and solvation energies etc. The horizontal axis is said to represent some sort of conformational entropy of the molecule. Each conformation is represented by a point on a multidimensional energy surface and (the most critical aspect): Conformations that are similar geometrically are close to one another on the energy landscape. Local maxima correspond to unfavorable high-energy conformations, valleys to more favorable low-energy conformations. In fact, the notion of entropy on the horizontal axis is a problem since no rigorous definition is given that would allow one to compute it actually.

The first investigations suggesting folding funnels were made by Bryngelson and Wolynes [16] in the late 1980ies. They explored the bumpiness of protein folding landscapes in simplified spin-glass models. Later, Leopold et al. [92] were the first who described in some detail how the shape of the folding funnel depends on the amino acid sequence by enumerating lattice heteropolymer conformations. Within the folding funnel view, a protein is supposed to change its conformation in ways that cause its energy to decrease. At the same time, the protein is of course subject to Brownian motion and it is thus constantly converted into different conformations. Uphill steps must also be taken into account. As mentioned before, the lateral expansion of a folding funnel at a certain energy level represents its conformational entropy. According to funnel theory, this is consistent with the assumption that the progress toward lower free energy conformations is accompanied with diminished conformational freedom - finally resulting in the native structure.

The ability of certain sequences to fold (or not to fold) to a native state has often been addressed in terms of the topology of the underlying energy landscape. Within this context, one must differentiate between random and protein-like heteropolymers. Random heteropolymers do not have a well-defined three-dimensional 'native' state, but a collection of completely different low-energy structures. To illustrate this, imagine a reaction coordinate Q (defined e.g. as the fraction of native tertiary contacts). It was proposed [118] that an ideally designed folding sequence should have the "energy of its conformations proportional to Q plus some roughness introduced by nonnative contacts". All stabilizing contacts should be equally distributed throughout the structure - the system is said to be "unfrustrated". This correlation between energy and structure should on the one hand favor the native conformation, one the other hand proportionally bias all non-native conformations and thus be responsible for the funnel shape of the landscape. In contrast to that, a random sequence would not exhibit such correlation, leading to a rough landscape (figure 24).

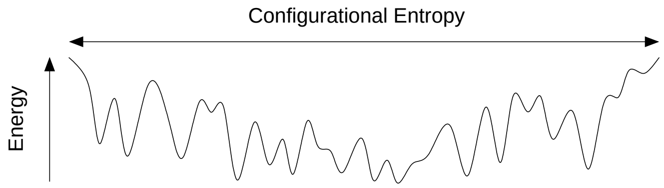


Figure 24: Energy landscape of a random heteropolymer. Different low-energy states are not only possible, but may also exhibit completely different structures with different sets of frustrated contacts.

The idea that real proteins need to minimize frustration was first proposed by Bryngelson and Wolynes [16]. Due to them, fast folding would be impossible on entirely rugged (frustrated) landscapes. Fast folding should only be possible because of guiding forces that stabilize native interactions in way more than one would expect by chance. Fast folding proteins are supposed to satisfy a principle of minimal frustration. The same authors were also the first who gave a hint that protein folding should be seen in terms of certain temperatures: A *folding temperature* T_f, below which the lowest energy-state is supposed to be stable. (Formally, folding temperature has been defined as the temperature where the native state is occupied 50% of the time.) Protein folding landscapes were known not no be perfect funnels. Due to this ruggedness, another temperature below which the kinetics is controlled by "long-lived low-energy traps" (and not a straight bias toward a native conformation) was proposed. This temperature was termed *glass transition temperature* T_g. Later, Socci and Onuchic [140] provided an operational definition of T_g. Given that trapping were not a problem, lowering the temperature should speed up folding because it favors collapse. Nevertheless, as the temperature gets lowered, there is a point where a rapid slowdown of folding happens. This temperature was called *kinetic glass transition temperature* and is similar to the *thermodynamic glass transition temperature* proposed by Bryngelson and Wolynes.

Much effort was put into investigation of foldability of protein-like heteropolymers throughout the 1990ies, mainly using minimalist models (section 2.5). Camacho and Thirumalai [18] studied kinetics of three different types of interaction potentials using a two-dimensional lattice system with relatively short chain lengths. They found both T_f and T_g as predicted by Bryngelson and Wolynes and argued that proteins in the region $T_f \leq T \leq T_g$ may correspond to a *molten globule* containing the bulk of the backbone native structure

Socci an Onuchic [140] investigated several different sequences of length 27 within a maximally compact $3 \times 3 \times 3$ cube on a simple cubic lattice. Studying kinetics of collapse and folding, they found that folding time depends on the sequence (as one would expect) and is related to the amount of energetic frustration in the native state. Collapse times - in contrast - turned out to be sequence independent within their model. The authors were able to identify two classes of sequences: good folders with $T_f > T_g$ and non-folders with $T_f < T_g$.

At this point is seems fair to formulate some critical remarks on the concept of folding funnels: First, it is not evident that proteins really do have unique ground states. A prominent example for this is the existence of chaperons, proteins whose function is to assist other proteins in achieving proper folding (see e.g. [158]). Originally detected as heat shock proteins, they play an important role in protein-protein interactions such as folding and assisting in the establishment of proper protein configuration and prevention of unwanted protein aggregation. Prions [123] (short for *proteinaceous infectious particle*) are another group of - pathologically relevant - proteins that contradict the assumption of unique ground states. Prion proteins can occur in different conformations and their distorted form has the ability to induce the "normal" form to become distorted. Although the exact mechanisms of action of these infectious self-reproducing protein structures is not yet known, it is now commonly accepted that they are responsible for a number of diseases generally classified under transmissible spongiform encephalopathy (TSEs) diseases, such as scrapie (a disease of sheep) and bovine spongiform encephalopathy (mad cow disease). These diseases affect the structure of brain tissue and are all fatal and untreatable. There is strong indication that the lethal human Creutzfeldt-Jakob Disease is caused by transmissible prions.

We mentioned earlier that a rigorous mathematical characterization of folding funnels has not been given yet. However, we can state that the concept - in the way it has been proposed in literature so far - seems insufficient. The fact that some proteins have the ability to refold from one structure to another and thus change their function gives rise to the assumption that energy barriers are indeed necessary. In other words we could say that folding funnels do exist, but it is not evident that they really look like the one shown in figure 23. Remember the vague definition of the horizontal axis, and the argument that 'drive to collapse should also be a drive towards a reduced ensemble of conformations'. One possibility to set up a reasonable scale for the lateral expansion of the folding funnel would be to define some "conformational entropy" as a function of the system's inner energy, i.e. $S_{conf} = k \ \ln N[E]$ with $N[E]$ being a measure for the density of states. However, even if this definition would be appropriate, it is not clear whether the protein moves upwards or downwards on this (S, E) surface. Another point that must be considered is the question whether geometrically similar structures are adjacent to each other in such a high dimensional funnel. As a result, anyone could argue that the native state of a biopolymer is a certain point on a hypersphere representing the energy landscape.

Much effort was put into investigations of protein folding funnels over the last years. Nevertheless, the concept is still insufficient for predicting the exact folding behavior or describing biopolymer dynamics qualitatively. On the other hand, it is appropriate as a conceptual model to get a principal impression on molecular driving forces.

For the purposes of this thesis we will rely on barrier trees, having a profound mathematical foundation (given in the last chapter) rather than folding funnels. We believe that barrier trees embody all the relevant quantitative information about the multi-valley structure of an energy landscape and should thus be regarded as a reasonable representation of the energy landscape.

6 Low-energy states of the energy landscape

We know from previous sections that it is necessary to have a list of *all* structures a given sequence can fold into within a certain energy interval in order to apply the algorithm presented in section 4.2. It was mentioned earlier that the situation is different in the case of lattice proteins from the RNA case. Lattice protein folding was shown to be NP-complete and hence there is no efficient algorithm available to determine a lattice polymer's ground state. It is further not possible to recursively calculate a set of suboptimal lattice protein structures, as it is possible for RNA. To escape the problem, we implemented a tool for exhaustive enumeration of all SAWs of given length on a certain lattice (`latticeSub`, section 2.8).

A different approach to generate the low-energy portion of an energy landscape was motivated by the poor results of the exhaustive enumeration technique obtained for longer chains on 3D lattices (though the tool is readily applicable to 2D lattices as well). The main idea is simple: Start from a well-defined structure on a certain lattice (if one wanted to investigate the lower part of the energy landscape one would of course choose a low-energy or near-optimal structure, if available) and, according to a pre-defined move set (section 3.1.2), generate all neighbor structures. Then take the just generated neighbors and generate their neighbors and so on. This procedure is repeated until a predefined amount of structures has been found. More formally, we can select S^0 as a start structure. The first step is to generate all its neighbors $\mathcal{N}(S^0) = (S^0_1, S^0_2, S^0_3, ..., S^0_n)$. An optional constraint would be to define a certain energy threshold for the neighbor structures. Any structure from $\mathcal{N}(S^0)$ that has an energy lower than the threshold is then a) written into a hash and b) a pointer to the just inserted hash entry is added to a list of hash pointers. The next step of the algorithm is to process the elements of the hash-pointer list. Similar to the initial step, all neighbors of $S^1 = S^0_1$, $\mathcal{N}(S^1) = (S^1_1, S^1_2, S^1_3, ..., S^1_m)$ are calculated. After that, $\mathcal{N}(S^1)$ is processed and (given that its energy is below the threshold), each structure is looked up in the hash if it has been seen before. If this is true, the structure is thrown away and the next structure is processed. If the structure has not been seen before, it is inserted into the hash and a pointer to the entry is put at the end of the hash pointer list. After all structures from $\mathcal{N}(S^1)$ have been processed, the hash pointer list-entry for S^1 is deleted and the procedure is repeated with

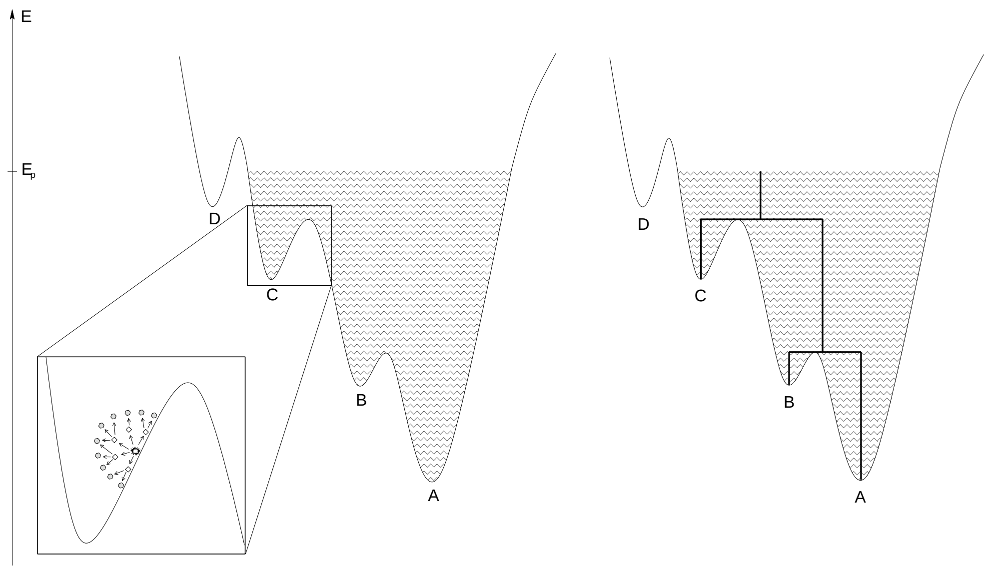


Figure 25: Schematic representation of the `latticeFlooder` algorithm (left plot). Starting from a certain conformation, all neighbor conformations are calculated repeatedly until all conformations in a certain region of the energy landscape are found (note that the shape of the energy landscape is not known at the beginning of the algorithm). Within this schematic example, all conformations with an energy lower than E_p are generated. Starting from a conformation in basin C, all conformations in basins B and A are found (illustrated by the schematic barrier tree in the right plot). Basin D is not found with this method since its energy is too high and it is thus not connected.

the last entry from the hash pointer list[13] as initial structure. The end of the algorithm is reached as soon as a) a predefined amount of structures has been found (limited by the size of RAM, currently approximately 40 million structures can be generated on machines with 4GB RAM) or b) all structures that are "reachable" from a distinct start-structure (constrained to an energy threshold) are found.

Figure 26 shows the results of some benchmarking (`SQ` lattice and **HPNX** alphabet) we did to find out about time efficiency of the two tools. Evidently, `latticeSub` is significantly faster for short sequences since its exhaustive search routine is implemented as a recursion algorithm. So if we wanted to generate a list of all SAWs of a short lattice protein, we would use `latticeSub`. On the other hand, `latticeFlooder` has a major advantage: This tool allows us in prin-

[13]From the computational point of view, it does not make any difference with respect to time efficiency whether to take the first, the last or any intermediate entry from the hash pointer list as next start structure.

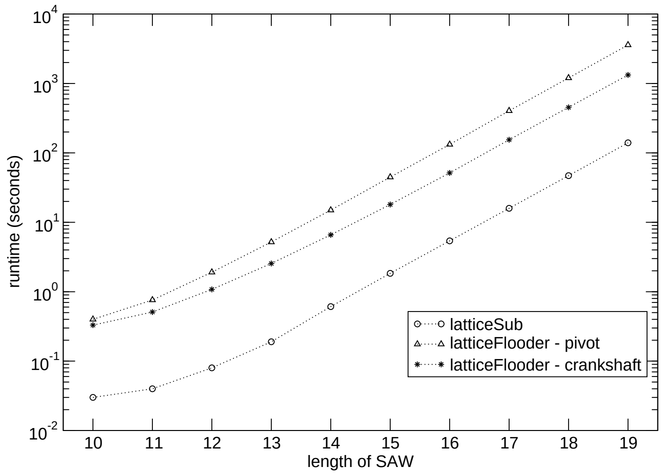


Figure 26: Benchmark `latticeSub` vs. `latticeFlooder` of variable-length **HPNX** alphabets of the `SQ` lattice. `latticeSub` is faster for short chains, `latticeFlooder` allows for a move set-specific calculation of structures and examination of longer SAWs.

ciple to generate the whole low-energy portion of some conformation space X - even for longer sequences, at the cost of more memory requirements. Another advantage of `latticeFlooder` is that this tool can generate neighbor structures according to a certain move set.

A typical application of `latticeFlooder` would be investigation of the often referenced 27-mer on the `SC` lattice. Assuming a poly-H sequence, the ground state is given by a $3 \times 3 \times 3$ cube with 28 HH-contacts. However, there exist 103346 compact $3 \times 3 \times 3$ structures (see e.g. [42]) and it would thus be inefficient to investigate this problem with `latticeSub`. `latticeFlooder`, on the other side, can generate a representative low-energy part of the energy landscape when starting form a compact $3 \times 3 \times 3$ conformation.

Figure 26 shows two series of `latticeFlooder` results, one for each move set we implemented (section 3.1). Since the hashing routines in `latticeFlooder` and `barriers` are the same, combined usage of these two tools yields not only *connected* barrier trees (section 4.1) but it is also assured that the whole list of structures can be handled computationally within a `barriers` run. In other words, if `latticeFlooder` can handle some amount of structures, `barriers` can handle it as well if the calculations were performed on the same machine.

7 Dynamics of Biopolymers

We developed the framework for our further investigation of the dynamics of biopolymer folding in the last sections. At this point, we have a reasonable representation of the energy landscape at hand (barrier trees) and the next step is to predict the folding behavior of biopolymers by numerical integration. Originally developed for investigation of RNA folding kinetics (and published recently [162]), this concept has been expanded to lattice proteins within the context of this thesis. Before illustrating the coarse-grained description using the barrier tree approach, we will illustrate our motivation by giving a short review of previous work in the field of RNA folding kinetics and present the underlying model.

7.1 The Model

A valuable method for investigating time evolution of RNA secondary structures was given by Flamm et. al. [47]. In this contribution, it has been shown that a good approximation to the few available quantitative and qualitative data[14] on RNA folding kinetics is obtained by modeling the conformational changes in terms of elementary steps of opening and closing of base pairs. For the purposes of this thesis, the tool has been adapted to lattice protein folding as well - `pinfold` simulates kinetic folding of lattice proteins in terms of elementary steps. The tools `kinfold` (RNA) and `pinfold` (lattice proteins) are capable of simulating the whole kinetic folding process of RNA / lattice protein molecules using the following model:

Let I be a sequence which specifies a set of structures with which it is compatible,

$$\mathcal{S}(I) = \{x_0, x_1, ..., x_m\} \cup \{0\} \tag{24}$$

where x_0 is the minimum free energy (mfe) conformation, $x_1..x_m$ are energetically ordered suboptimal conformations and 0 is the denatured, open chain conformation. The set $\mathcal{S}(I)$ and the move set introduced in section 3.1.1 form the conformation space. A trajectory $\mathcal{T}(I)$ (as computed by `kinfold`) is a time-ordered series of secondary structures in $\mathcal{S}(I)$. Because the conformation space of secondary structures is always finite, every trajectory will reach x_0 after sufficiently

[14] In fact, very few experimental data are available at present to estimate transition rates between different RNA secondary structures.

long time. The *folding time* τ (associated with a trajectory) is defined as the first passage time, that is, the time elapsed until S_0 is encountered first. Due to the fact that τ may well be too long for a computer simulation, one can distinguish between trajectories that actually attain the ground state within the limits of a simulation from those that are trapped in a thermodynamically suboptimal conformation.

Translated into the language of chemical kinetics, the system is the biopolymer chain and a state of the system is a certain conformation of the chain. Given the move set, biopolymer folding can then be modeled as a *Markov process* in conformation space. To do so, it is necessary to introduce a *transition rate* k_{yx} between two distinct states x and y, which is a small nonnegative real number that determines how the probability of the transition from x to y increases with time. The probability distribution P of structures as a function of time is ruled by a set of forward equations, also known as the master equation

$$\frac{dP_t(x)}{dt} = \sum_{y \neq x} [P_t(y)k_{xy} - P_t(x)k_{yx}]. \tag{25}$$

Within this stochastic formulation, $k_{yx}\Delta t$ is the probability that a transition from a distinct state x to another distinct state y occurs within the infinitesimal time interval Δt. For the solution of the last equation (in matrical form), it is necessary to formulate a square intensity matrix (transition matrix) $\mathbf{R} = (r_{xy})$ which contains the transition rates between different states of the system

$$r_{xy} = k_{xy} \qquad \text{if } x \neq y \tag{26}$$

$$r_{xx} = -\sum_{y \neq x} k_{yx} \qquad \text{otherwise} \tag{27}$$

Thus, the master equation (i.e. the probability that the molecule has the secondary structure x at time t) can be written as

$$\frac{dP_t(x)}{dt} = \sum_{y} P_t(y)r_{xy} \tag{28}$$

or rewritten in matrix form:

$$\frac{d}{dt}P_t = \mathbf{R}P_t \tag{29}$$

In principle, equation (29) can be integrated numerically. Tacker et al. [145] used this technique to assess the feasibility of particular folding pathways of melting and refolding of $\mathrm{tRNA}^{\mathrm{phe}}$. Breton et al. [13] proposed a rigorous model of a sequential RNA folding process during transcription using this ansatz.

We are interested in calculating the temporal distribution vector P_t, which can be calculated from the explicit solution of (29)

$$P_t = e^{t\mathbf{R}} P_0 \tag{30}$$

where P_0 is the initial distribution vector.

A fundamental requirement of this model is the concept of *detailed balance*, i.e. the microscopic fluxes in one direction must equal the microscopic rates in the other direction. In other words, microscopic reversibility must be guaranteed and there exists a unique probability distribution of the Markov chain satisfying the *balance equations*

$$\pi_y = \sum_x r_{yx} \pi_x \tag{31}$$

for all y.

What still needs to be established is a rule for the transition rates r_{xy} between *neighboring* structures. The transition state model dictates an expression of the form

$$r_{yx} = r_0 e^{-\frac{E^{\neq}_{yx} - E(x)}{RT}} \quad \text{for } x \neq y \tag{32}$$

where the transition state energies $E^{\neq}_{yx}$ must be symmetric to assure detailed balance, $E^{\neq}_{yx} = E^{\neq}_{xy}$. In the simplest case one can use

$$E^{\neq}_{yx} = \max\{E(x), E(y)\} \tag{33}$$

which amounts to the Metropolis rule of simulated annealing. The parameter r_0 could be used to gauge the time axis from experimental data, in the following we simply use $r_0 = 1$.

Other models for the transition rate between two states are possible as long as detailed balance is satisfied. According to Kawasaki [84], the symmetric rule evaluating the transition between the two states x and y connected by the reaction channel α is formulated as:

$$k_{yx} := e^{-\frac{E^{\neq}_{yx} - E(x)}{2RT}} \tag{34}$$

Note that the free energy difference ΔE between the two states x and y must be divided by $2RT$ to get the detailed balance right. The Kawasaki dynamics approaches the Boltzmann distribution at equilibrium because it satisfies microscopic reversibility [69]. For a detailed discussion of possibilities to formulate the transition probabilities in the lattice protein case, see [26].

7.2 Barrier Tree Kinetics

In the last section, the general model for the kinetic folding of biomolecules was presented with the RNA example. In a previous section it was shown that the conformation space grows exponentially with the chain length of the biopolymer. Due to the fact that the algorithm of `kinfold`/`pinfold` makes use of a stochastic model, a great many trajectories have to be calculated to get a representative impression on the real folding behavior of the molecule. Furthermore one has to bear in mind that *all* legal structures of a biomolecule within a certain energy interval must be considered in such a simulation. Thus, a realistic description of the energy landscape or the dynamics of biopolymers based on all configurations is very intensive in terms of time and computer resources. For RNA, this means that it is not possible to simulate the kinetic folding of sequences with $n > 500$. The situation is even worse for the lattice protein case. As a matter of fact it is necessary to replace this stochastic model with a deterministic one: We need to coarse-grain the representation of the energy landscape. But what states should be considered within the new, restricted conformation space? A short investigation leads us back to the concept of barrier trees. To be more precise, we will map the original (huge) conformation space onto the barrier tree and state that such a tree represents the energy landscape 'as-is'. Basins and saddle points in the tree correspond to basins and saddle points of the folding landscape, respectively (see below). With this model, it is interesting to find out about the population probability of certain local minima on the barrier tree with respect to the fact that they are separated by more or less high energy barriers. In fact, we focus our investigations on the following questions:

- When starting the simulation at a specific local minimum of the tree (e.g. the denatured, open chain conformation), how long does it take for the system to reach an equilibrium state?
- To which extent are other local minima being populated on the way from the start structure to the minimum free energy structure?

- What are first passage times of specific local minima?

To investigate these questions we model the dynamics of a biopolymer as a continuous time Markov chain. As mentioned before, the conformation space is reduced in a way that we are only interested in local minima present in the barrier tree. In our special case, the *system* is the biopolymer chain and a *state* is the population probability of basins of the energy landscape.

Let $\mathbf{\Pi}$ be a partition of the state space X. For $\alpha, \beta \in \mathbf{\Pi}$ define $f[\alpha, \beta] = \min_{x\in\alpha} \min_{y\in\beta} f[x, y]$, i.e. $f[\alpha, \beta]$ is the minimal saddle point energy connecting between two points in the two different classes α and β of $\mathbf{\Pi}$. It follows that $f[\alpha, \alpha] = \min_{x\in\alpha} f(x)$.

Definition 22. *The partition* $\mathbf{\Pi}$ *is*

> **compatible** *with the energy landscape* f *if for all* $x \in \alpha$ *and* $y \in \beta$ *with* $f(x), f(y) \le f[x, y]$ *holds* $x \xleftrightarrow{f[x,y]} y$.
>
> **strictly compatible** *with* f *if it is compatible and for all* $\alpha, \beta \in \mathbf{\Pi}$ *there is* $\eta_{\alpha\beta} < f[\alpha, \beta]$ *such that* $x, x' \in \alpha$, $f(x), f(x') < f[\alpha, \beta]$ *implies* $x \xleftrightarrow{\eta_{\alpha\beta}} x'$.

Compatibility implies that the level sets of α and β are connected at the level of their mutual saddle point energy, strict compatibility requires that the classes of $\mathbf{\Pi}$ remain connected just below the saddle point energies. Trivially, the discrete partition $\mathbf{\Pi}_0 = \{\{x\} \,|\, x \in X\}$ is strictly compatible with f. A non-trivial example are the gradient basins.

Lemma 23. *Suppose* f *has unique gradient basins. Then the partition* $\mathbf{\Pi}_\gamma$ *consisting of gradient basins is strictly compatible with* f*.*

Proof. Let $x, y \in G(u)$. Then $\gamma^\infty(x) = \gamma^\infty(y) = u$, i.e. there are monotonically decreasing paths connecting x and y ending in u. Therefore $x \xleftrightarrow{\max(f(x),f(y))} y$. Let α, β be two gadient basins. Denote the corresponding local minima by u_α and u_β. Then $u_\alpha \xleftrightarrow{E[\alpha,\beta]} u_\beta$ by definition. For $x \in \alpha$ with $f(x) < E[\alpha, \beta]$ and $y \in \beta$ with $f(y) < E[\alpha, \beta]$ we have therefore $x \xleftrightarrow{f(x)} u_\alpha \xleftrightarrow{E[\alpha,\beta]} u_\beta \xleftrightarrow{f(y)} y$ and hence $x \xleftrightarrow{E[\alpha,\beta]} y$. Thus $\mathbf{\Pi}_\gamma$ is compatible with f. Furthermore, set $\eta_{\alpha\beta} = \max\{f(x) \,|\, x \in \alpha \cup \beta \text{ and } f(x) < E[\alpha, \beta]\}$. We see immediately that $\eta_{\alpha\beta} < E[\alpha, \beta]$ and $x \xleftrightarrow{\eta_{\alpha\beta}} x'$ for all $x, x' \in \alpha$ with $f(x), f(x') < E[\alpha, \beta]$. Thus $\mathbf{\Pi}_\gamma$ is strictly compatible with f. □

We call the classes of a compatible partition *macrostates*.

Consider the partition of X defined by the gradient basins $\mathcal{B}(z)$ of the local energy minima (section 3.2). To each macrostate α we can assign the partition function

$$Z_\alpha = \sum_{x\in\alpha} e^{-E(x)/RT} \tag{35}$$

and the corresponding free energy

$$G(\alpha) = -RT \ln Z_\alpha \tag{36}$$

Let us now turn to the transitions between macrostates. Suppose we know the transition rates r_{yx} from x to y. Then

$$r_{\beta\alpha} = \sum_{y\in\beta}\sum_{x\in\alpha} r_{yx}\mathrm{Prob}[x|\alpha] \quad \text{for } \alpha \neq \beta \tag{37}$$

where $\mathrm{Prob}[x|\alpha]$ is the probability to occupy state $x \in \alpha$ given that we know that the process is in macrostate α. The kinetics of the molecule in terms of its macrostates is given by the master equation

$$\frac{\mathrm{d}p_\alpha}{\mathrm{d}t} = \sum_{\beta\in\mathbf{\Pi}} r_{\alpha\beta}p_\beta(t) \tag{38}$$

where $p_\alpha(t) = \sum_{x\in\alpha} p_x(t)$ and $r_{\alpha\alpha} = -\sum_{\beta\neq\alpha} r_{\alpha\beta}$. Assuming (local) equilibrium we have $\mathrm{Prob}[x|\alpha] = e^{-E(x)/RT}/Z_\alpha$ and hence

$$r_{\beta\alpha} = \frac{1}{Z_\alpha}\sum_{y\in\beta}\sum_{x\in\alpha} r_{yx}e^{-E(x)/RT} \tag{39}$$

The point here is that we can compute $r_{\beta\alpha}$ "on flight" while executing the `barriers` program if two conditions are satisfied: (a) For each x we can efficiently determine to which macrostate it belongs and (b) the double sum in Eq. (39) needs to be evaluated only for neighboring conformations $(x, y) \in \mathfrak{M}$. Condition (b) is obviously satisfied in the landscape model since $r_{yx} = 0$ by definition unless x and y are neighbors.

Condition (a) is easily satisfied for each of the gradient basins: in each step of the `barriers` algorithm all neighbors y of the newly added structures x that have a smaller energy have already been processed. Hence, if their assignment to a gradient basin is known, the assignment for x equals the one for its lowest energy neighbor. Initially, each local optimum forms the nucleus of new gradient basin, hence the macrostate to which x belongs can be determined in $\mathcal{O}(\delta)$ operations, where δ is the maximum number of neighbors of a secondary structure.

We can use the transition state model to define the free energies of the transition state $G^{\neq}_{\alpha\beta}$ by setting

$$r_{\beta\alpha} = r_0 e^{-\frac{G^{\neq}_{\beta\alpha} - G(\alpha)}{RT}} \tag{40}$$

A short computation then yields

$$G^{\neq}_{\beta\alpha} = -RT \ln \sum_{y\in\beta} \sum_{x\in\alpha} e^{-\frac{E^{\neq}_{xy}}{RT}} \tag{41}$$

as one would expect. This allows us to redraw the barrier tree (which was given in terms of the energies of meta-stable states and their connecting saddle points) in terms of free energies of the corresponding macrostates and their transition states. This approach will be denoted *rates* or *macrostate process* from now on.

Although the approximation just presented is suitable for routine calculations, further coarse-graining can be achieved. The simplest and most straightforward approximation for the folding dynamics is the Arrhenius law for transitions on the barrier tree. Transitions occur only between local minima that are directly connected by a saddle point, and the transition state energies are approximated by the saddle point energy $E[\alpha, \beta]$. If $\mathbf{\Pi}$ is the partition compatible with f then $E(\alpha) = \min_{x\in\alpha} f(x)$. Hence we can derive

$$\tilde{r}_{\beta\alpha} = e^{-\frac{E[\alpha,\beta] - E(\alpha)}{RT}} \tag{42}$$

for the rates between macrostates α and β. This approximation (which we will call the "*tree process*" from now on) completely neglects entropic terms that arise because there are many possible paths connecting two local minima. The rates process can thus be viewed as "tree process plus activation entropies".

At this point it seems fair to say some words on the limitations inherent in the just presented model: We assume that the dynamics of a folding biopolymer can be modeled as dynamics on a highly simplified, coarse-grained state space. It is necessary to mention that - in contrast to our model - the conformation space of real biomolecules *in vivo* is not limited to some 'macrostates' (represented by the local minima of the landscape), i.e. the dynamics develops by making use of a lot of states. This leads us to the question: What *is* the actual dynamic behavior of a biomolecule and how can we prove the correctness of our simulations? A straightforward answer to that question would be to consider the whole conformation space of the observed molecule within our simulations. It was shown in section 3.1 that the number of structures X grows exponentially with sequence length. Even for small molecules it becomes very soon very big, even as big as it cannot be treated any more within a computer simulation. The limiting factor concerning computer resources is RAM, as the transition matrix (section 7.1) has to be stored as a whole during diagonalization. Nevertheless it is possible to calculate the dynamic behavior for some reasonably small conformation spaces with up to a few thousand structures on modern machines with reasonable amount of RAM. We call this the *full process* - in contrast to the *Arrhenius (tree) process* on the one and the *rates process* on the other hand within our model. Due to the fact that the full process includes the entire conformation space of a given biomolecule, it represents the 'real' dynamic behavior of the sequence and hence is an ideal reference for our simulations. We modified `barriers` to gather information on the neighborhood relations among all configurations. Within the *full process* it is possible to formulate transition rates between the different structures using the Metropolis and the Kawasaki rule introduced in section 7.1.

The main problem is the calculation of population densities in terms of macrostates of the state space, i.e. local minima of the barrier tree. The structure probability distribution for the allowed local minima of the barrier tree can be calculated recursively from equation 30. Unfortunately, $\mathbf{R}$ is a matrix of dimension n where n is the number of local minima treated in the current simulation. Since it is very difficult and inefficient to evaluate an expression like $e^{\mathbf{R}}$, similar to the right side of equation (30), the calculations are performed in the eigen space of the system. The transition matrix $\mathbf{R}$ is non-symmetric by definition and it is necessary to symmetrize it by multiplying it with the equilibrium distribution vector π from the left side and $\pi^{-1/2}$ from the right side before applying a diagonalization algorithm to it (see [161] for details). However, this is only possible when we are interested in the equilibrium distribution of the Markov chain. The situation is different when we are instead interested in *refolding* times (first passage times) of certain states. In that case, *one* state is absorbing and it is thus impossible to symmetrize the original transition matrix. Consequently, the usual diagonalization routines for symmetric matrices cannot be applied in that case. Instead, we use Schur decomposition [57] to diagonalize the non-symmetric transition matrix.

Another point that has to be mentioned within this context is the fact that within `kinfold/pinfold` simulations, we always concentrate on specific configurations of a biomolecule. We are not interested in the equilibrium solution of the master equation 25, but rather in computing the distribution of first passage times from some initial state to the thermodynamic ground state. In other words, we define certain start structures and are interested in the time that elapses until a specific stop structure is reached. In this framework, the first passage time of course represents the folding time. Within the reduced description of the folding process, we do not have specific structures any more, instead we deal with macrostates. This implies a problem with direct comparison of `kinfold` and `treekin` simulations. The solution is achieved by introducing an absorbing state Ω that is only accessible from the macrostate ω containing the stop structure u with a rate

$$r_{\Omega\omega} = r_0 e^{-E_u/RT}/Z_\omega \tag{43}$$

7.3 Computational Results

In the previous chapters the theoretical background as well as the underlying models of this thesis were introduced. With knowledge of the fundamental properties of biopolymer chains, the move set, the landscape described by barrier trees, and the deterministic formalism given in the last section we are now able to investigate the dynamic behavior of biopolymers of moderate size, i.e. this allows us to calculate the time-evolution of population probabilities of local minima on the barrier tree. We will give some examples of our calculations in this chapter - RNA dynamics on the one side as well as representative examples of lattice protein dynamics on the other side. The upcoming sections are separated in the following manner:

- We will give four examples of RNA dynamics. First a small molecule is used to show the principal behavior of kinetic folding of RNA. Then, a more sophisticated RNA chain with length $n = 20$ (and more appealing dynamics) as well as a RNA switch are examined. Finally, we demonstrate the capabilities of our tool with the well-known tRNA$^{\mathrm{phe}}$ sequence.
- The dynamics of lattice proteins is illustrated with different lattices. Beginning with a simple sequence on the `SQ` lattice we demonstrate kinetic

aspects of biopolymer folding with degenerate energy landscapes. We will further select a specific sequence and show the coarse-grained dynamics of this artificial biopolymer on different lattices compared to exact Monte Carlo simulations.

7.3.1 RNA Dynamics

As a first application of the algorithm we will analyze a short artificially designed RNA chain with sequence `UAUGCUGCGGCCUAGGC` (called lilly) and length $n = 17$. There are two reasons why we decided to chose this sequence: First, the whole conformation space X consists of only 810 secondary structures and second, although it has a very simple sequence, the molecule has two ground states with equal energy (-0.7 kcal/mol). Figure 27 shows the barrier tree of lilly, which gives an impression on the simple shape of the associated energy landscape[15]. There are 14 local minima in the barrier tree that are used as macro-states for our coarse-grained approach. The denatured, open-chain conformation is represented by local minimum 5 which is directly connected to local minimum 2 via an energy barrier of 2.1 kcal/mol.

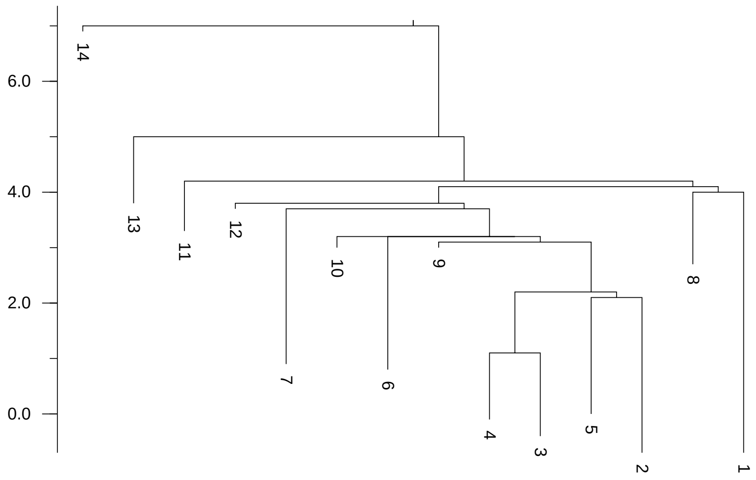


Figure 27: The barrier tree of the artificially designed RNA sequence lilly without Shift-moves. There are two ground states with equal energy (-0.7 kcal/mol). Local minimum 5 corresponds to the denatured, open-chain conformation.

[15] Note that all RNA-related examples shown here - apart from the third (RNA switch) - are calculated without 'Shift-moves' (section 3.1.1) due to computational efficiency.

The simplicity of this example allows us to directly integrate equation (25) and compare it with the coarse-grained dynamics. Figure 28 shows how the population densities p_α of the basins of attraction of some local minima α evolve with time. We did these calculations for the tree, the rates and the full process. Basin 5 was assigned a population probability of 1 in all three cases which means that 100 percent of the population is situated in basin 5 at the beginning of the simulation. The dynamics was simulated until an equilibrium population distribution was reached. In other words, we started the simulation from the open-chain conformation and let the system attain a stable thermodynamic equilibrium distribution.

The upper plot in figure 28 shows the dynamics using the Arrhenius (*tree*) model, the middle plot was calculated with the macrostate (*rates*) model and the lower plot - as reference - illustrates the *full* dynamics without simplification and coarse graining. For the full process, all 810 secondary structures of the conformation space were considered for the simulation. Common to all three plots is the general shape of the curves representing population probabilities of some local minima. Population of the 'start' basin 5 decreases rapidly and is at approximately 12-14 percent after 100 (arbitrary) time-steps. Beginning in the region of approximately one time-step, other basins are populated significantly, basin 2 has its population maximum in the region of approximately 50-100 time-steps with a probability between 34 and 38 percent. Within this time-frame also the ground state (thick curve in the plots) gets populated slightly. The final phase in time evolution of this short example sequence starts at approximately 100 time-steps and this is the region where A) the ground state is populated up to its equilibrium value of 28.5 percent and B) population probability of all other basins slightly decreases down to their equilibrium value. The tree process is slightly faster in reaching the equilibrium distribution (978 time units) than the rates process (2794 time units). The full process shows the slowest dynamics (6949 time-steps). The greater time consumption of the latter can be explained by the larger number of states that are considered within the full (exact) process. See the table below for the equilibrium probability distribution of the 6 lowest energy basins shown in figure 28. Note that the remaining 2.35 percent to reach a total population probability of 1 is shared among the remaining basins of the tree that are not shown in figure 28.

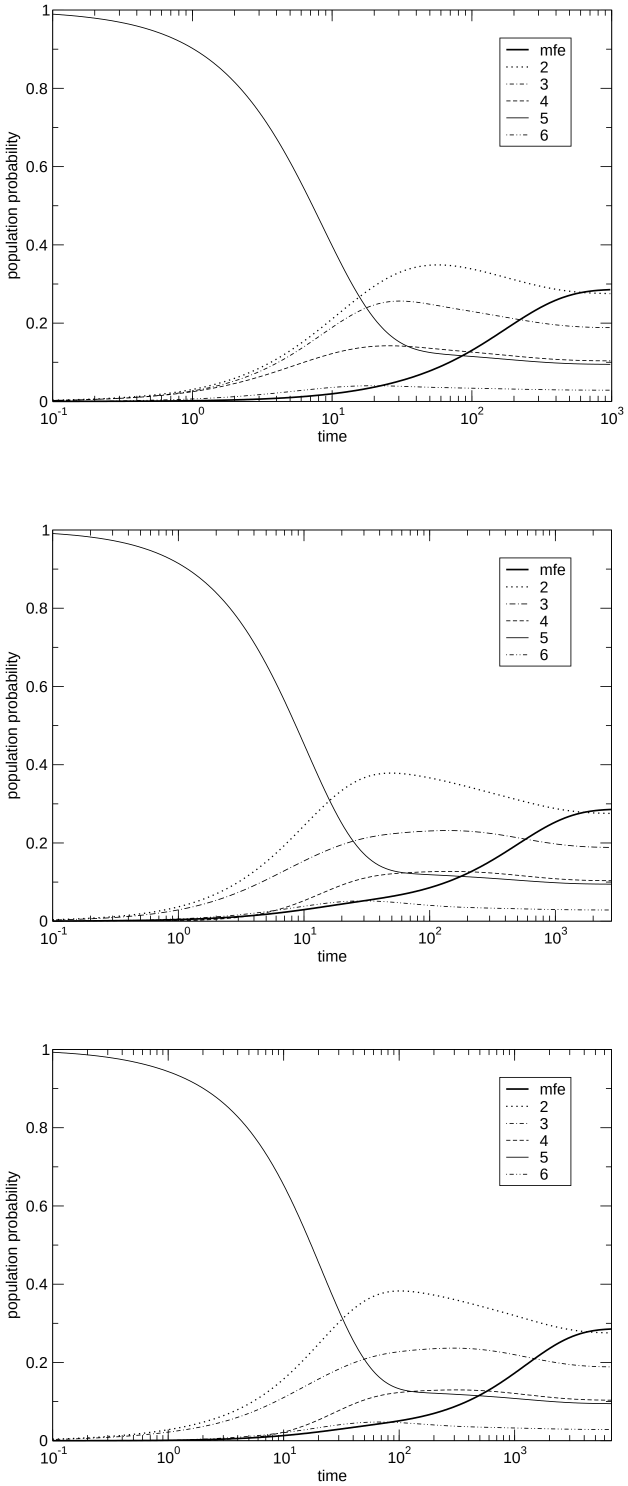


Figure 28: Dynamics of the short model RNA sequence lilly. **Upper plot:** tree process, **middle plot:** rates process, **lower plot:** full process. See text for details.

local minimum	population probability	secondary structure
1	0.2856	`...(((........)))`
2	0.2755	`.........((....))`
3	0.1885	`...((....))......`
4	0.1035	`((.((....)).))...`
5	0.0947	`.................`
6	0.0287	`....(((.....)))..`

Despite the simplicity of this sequence, it can be used as a representative example to demonstrate the capabilities of our algorithm and its variants. Figure 28 illustrates good qualitative accordance of the coarse-grained approach compared to the full process. The *rates* process yields somehow better results than the Arrhenius approach since it considers all microscopic rates between neighboring states in different basins and thus more closely resembles the real dynamics.

As a second example, we chose the sequence `CUGCGGCUUUGGCUCUAGCC` with length $n = 20$ and a conformation space consisting of 3886 secondary structures. Figure 29 shows the barrier tree of this artificial RNA molecule which we will denote xbix here. Neglecting shift-moves, the barrier tree has 34 local minima and the open chain conformation is represented by basin 8. As in the previous example, we used this macro-state as starting point for our simulation and let it run until convergence to the thermodynamic equilibrium distribution.

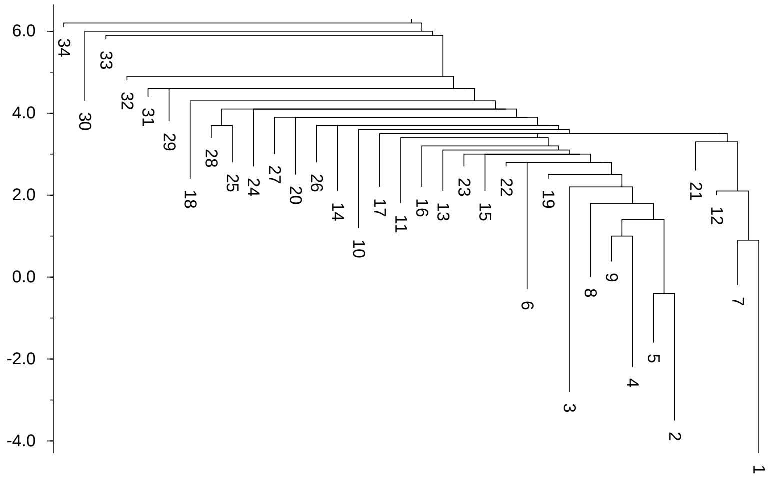


Figure 29: The barrier tree of the artificially designed RNA sequence xbix without Shift-moves.

Figure 30 shows the results of the simulation. Again, the moderate total number

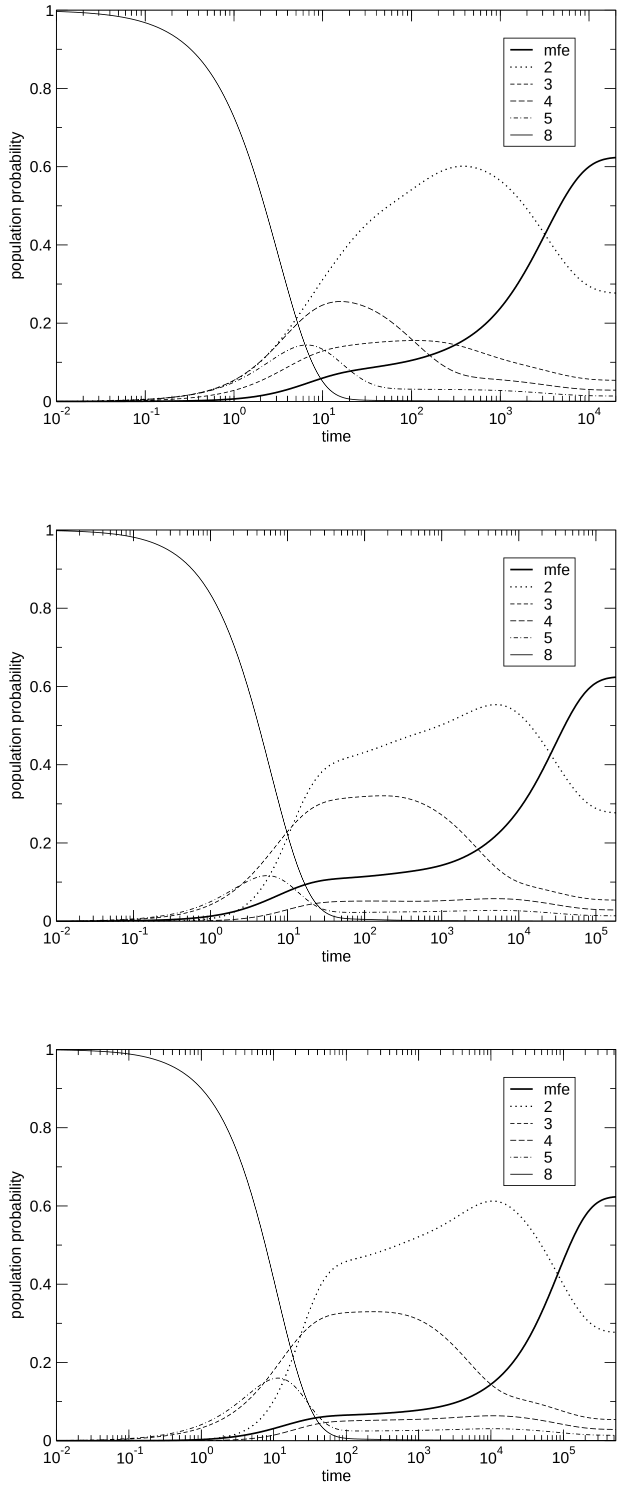


Figure 30: Dynamics of the artificial RNA sequence xbix. **Upper plot:** tree process, **middle plot:** rates process, **lower plot:** full process. See text for details.

of conformations allows for a direct integration of equation (25) of the microscopic process and direct comparison with the coarse-grained dynamics. At a first glimpse, all three plots in figure 30 exhibit qualitatively similar behavior. Just like in the simple example before, the macrostate approach as well as the full process tend towards a longer time-frame until the thermodynamic equilibrium distribution is reached. Nevertheless, all three approaches reach an equilibrium state within approximately the same time order of magnitude. Common to all three plots is the fact that the population density of the start basin has disappeared almost entirely after less than 100 time-steps. Within the tree process, local minima 4 and 5 (that are near neighbor states of local minimum 8) are populated at very early stages of the simulation. At the same time, also local minimum 2 (the direct father of 8) is populated. In contrast to the other approaches, local minimum 4 has a distinct population maximum at approximately 16 time-units and it is populated to a higher degree in the tree process than in the macrostate and the full process. The major difference among tree and macrostate/full approach is the shape of the curve associated with local minimum 2. From the upper plot in figure 30 we see that it is generally very smooth for the tree process and has a distinct shoulder at approximately 40 time-units in the rates as well as in the full process. We explain this by a different population probability of local minimum 3 in the range between 10 and 1000 time-units: This macrostate is populated more than twice as much in the rates/full process (approximately 33 percent) than in the tree process (approximately 15 percent) and thus gains population probability at the expense of local minimum 2. The final stage of the simulation is again common to all three processes. Basin 2 has a pronounced population maximum (390 time units for the tree approach, approximately 10^4 time-units for the macrostate/full approach) and loses part of this population in favor of the ground state. See the table below for a list of equilibrium probability values of the local minima shown in figure 30.

local minimum	population probability	secondary structure
1	0.6233	....((((........))))
2	0.2769	(((.(((....))).)))..
3	0.0540	..........(((....)))
4	0.0288	..(((((....)))...)).
5	0.0138	....(((....)))......
8	0.0005	....

The xbix example illustrates that there is excellent agreement between the macro-state approximation (middle plot in figure 30) and the full process (lower plot in figure 30). The Arrhenius law gives a qualitatively correct description of the process, although quantitative details are significantly different.

After illustrating the capabilities of our algorithm with two small artificial RNA sequences we will now turn to a RNA molecule with a) a very large conformation space and b) a very interesting behavior. To be more precise, we will focus our investigation on the bi-stable RNA switch lz04 with sequence `CAUCAUUUCAGCCGUAA-CCAUGAGAUGAUGGUUGCAACUAGUUCCCGUGAGGGAGUUUG` with $n = 59$. Bi-stable RNA molecules (also denoted RNA switches, see section 2.1) can fold into two or more thermodynamically stable secondary structures that are separated by a high energy barrier, which means that besides the subtree containing the global minimum, there are other dominating subtrees in the barrier tree. Figure 31 shows the 50 local minima with lowest energy of the energy landscape that were used as macro-states for the dynamics simulations. In our example, the two stable conformations are separated by an energy barrier of 13.4 kcal/mol. Note that in the left part of figure 31 there is another distinct energy barrier of 6.3 kcal/mol between the subtree containing local minima 40 and 46 with the subtree containing local minimum 2. We mention this because we used local minimum 46 as start point for the simulation (the denatured, open-chain conformation is not represented by any of the 50 minima in figure 31 because its energy is too high).

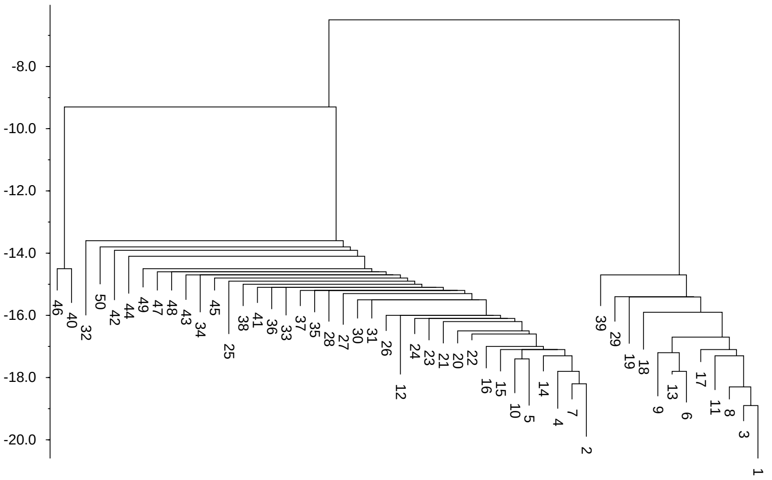


Figure 31: The barrier tree of an artificially designed RNA switch.

The large number of conformations X of this example sequence prohibits simulation of the full process without coarse graining. We rather show the macrostate dynamics in figure 32. The major difference compared to the previous simulations is that we are not interested in the time the system takes to reach a thermodynamically stable equilibrium distribution, but rather the *refolding* time from a selected start structure (local minimum 46 in the very left part of figure 31) to the ground state (global minimum 1 on the right of figure 31). The simulation starts with a rapid loss of population of local minimum 46 from 100 percent to approximately 32 percent within the first 4 time-steps. At the same time, local minimum 40 becomes populated and even after 1.7 time-steps both minima are equally populated at 50 percent. Shortly after, the dynamics shows an interesting behavior: The rapid rise of 40 and the rapid decline of 46 seem to stop and both remain populated in a quasi-stable state within the time-frame of 10 to 100 time-units. We find that the population is exclusively shared among these two local minima at times < 100. Looking at the barrier tree, this behavior becomes clear: 46 is separated from 40 only by a small barrier of 0.7 kcal/mol, whereas the barrier of 40 is 6.3 kcal/mol. So we can say that in the early stages

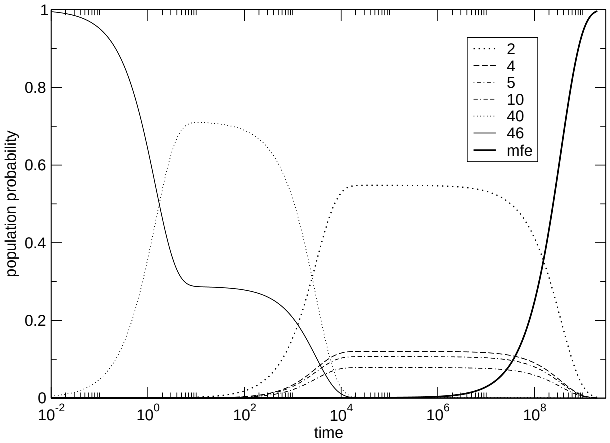


Figure 32: Macrostate Dynamics of the designed RNA switch lz04. The simulation is started at local minimum 46 and the ground state 1 is made absorbing. Local minima 40 and 2 are populated significantly during the simulation run. See text for details.

of the simulation, the population jumps back and forth between these two states. Starting at approximately 100 time-steps, the molecule overcomes the first large energy barrier and the way is open for population of energetically deeper local minima. Interestingly, the population distribution after some 20000 time-steps is completely different from that in the early stages of the simulation: 46 and 40 have completely lost their percentage of population in favor of the deepest local minima in the left subtree of figure 31. At this point, another quasi-stationary population distribution has established that represents the population dynamics in the (long) time-frame from 10^4 to 10^8 time-units.

By that time, local minimum 2 is populated almost at 55%, 4 at 12%, 10 at 10% and 5 almost at 8%. Other local minima that are not shown in figure 32 are also populated at rates of less than 5%. The quasi-stationary behavior of the molecule in this period is a direct consequence of the high energy barrier of local minimum 2 (13.4 kcal/mol). Finally, the molecule is able to overcome this high barrier and the ground state is populated starting at approximately 10^6 time-steps. The refolding process is finished with a 100% population of the absorbing ground state after 2×10^9 time steps.

Figure 33 shows the complete refolding path consisting of 64 steps from local minimum 46 to the ground state. The upper part of the image shows the energy profile. Basin 46 is located at the very right corner of the plot, the ground state at the left corner. Saddle points are labeled with a capital `S`, local minima on the path are labeled with a capital `L` and an associate number from `barriers`. Note that local minima numbers by far exceed the 50 lowest-energy minima from the barrier tree in figure 31 because the refolding makes it necessary to open favorable base-pairs. Starting from 46, local minimum 40 is visited first. After that, the molecule must climb up a first high energy barrier of 6 kcal/mol. As one would expect from the barrier tree, the refolding path includes local minimum 2 (after 13 elementary steps), which can be impressively seen as a deep valley in the energy profile. After visiting the meta-stable structure, the molecule has to overcome an ever higher energy barrier of 10.7 kcal/mol to escape the big valley and first reach local minimum 400 after 22 steps. Several other high-energy minima are visited afterwards on the path towards the ground state, the structure with highest energy is the saddle between local minima 490 and 232 with an energy of -6.5 kcal/mol (after 32 elementary steps). The large number of unfavorable high energy-intermediates can be explained by the fact that the nucleation region started at step 24 is not yet optimal. Finally, after the high-energy saddle between local minima 271 and 170 has been visited, the way is open for formation of more favorable structures, finally resulting in the minimum free energy structure. The whole refolding path and associated energies are given in Appendix C for reference.

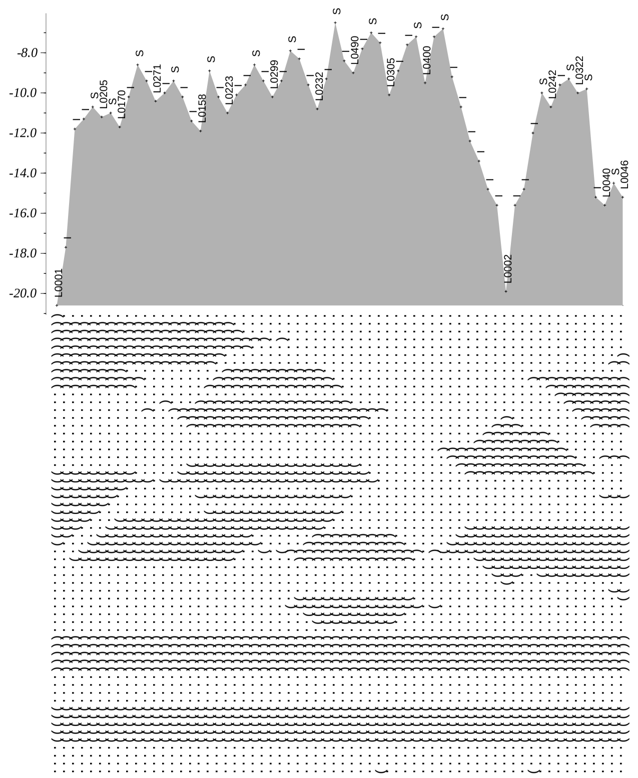


Figure 33: Energy profile of the complete refolding path from local minimum 46 (r.h.r) to the minimum free energy structure (l.h.s). The lower part of the plot lists involved secondary structures. Letter code in the upper part: `L`: Local minimum (along with its number as calculated by `barriers`). `S`: Saddle point. Dashes and dots represent intermediate structures.

The last RNA example that we will use here to demonstrate our algorithm is the well-known yeast $tRNA^{phe}$ sequence. It has a length of 76 nucleotides and some 2.8×10^{17} possible secondary structures. To recover all saddle points between energetically low local minima we considered the approximately 25 million structures within 15kcal/mol of the ground state, and used `barriers` to compute the 1000 lowest energy local minima as well as the rates between the corresponding macro-states. Only minima with a depth of at least 1kcal/mol were considered in the process. Figure 34 shows the 100 lowest energy local minima of the energy landscape of this tRNA.

Obviously, solving the master equation of the full process including the dynamics of all allowed secondary structures is out of the question for such a large conformation space, just like in the RNA switch example before. Instead we compare our coarse grained dynamics to a stochastic sample of trajectories generated by

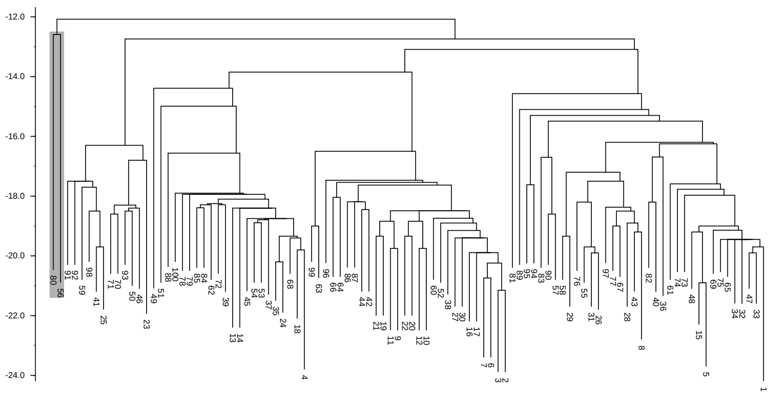


Figure 34: The barrier tree of tRNA$^{\text{phe}}$. Only the 100 lowest energy local minima are shown, local minima 56 and 80 are the two left-most states (highlighted in gray).

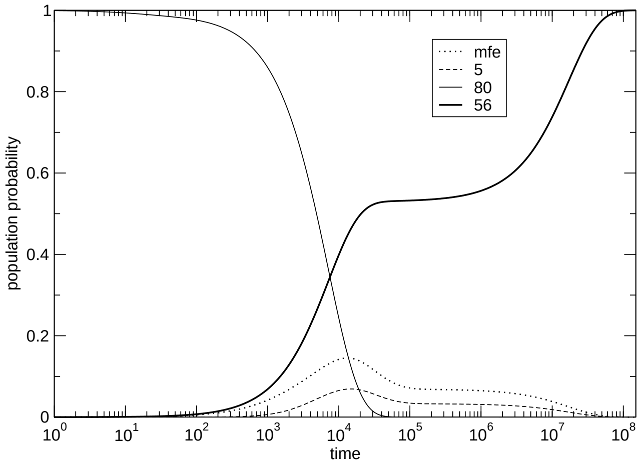


Figure 35: Refolding of a tRNA molecule: Macrostate simulation of the refolding event starting at 80 and absorbing state attached to basin 56. Two different folding pathways are indicated by the plateau of the curve associated with basin 56.

`kinfold` (see section 7.1). Computing the occupancy of each macro-state from `kinfold` trajectories is very expensive in terms of computer resources, in particular because the time to equilibration becomes too long. Instead we have used `kinfold` to compute first passage times by defining a stop structure in addition to the start point of each trajectory. In order to compare the results from `kinfold` with those of the macrostate process, we introduced the previously mentioned additional absorbing state (equation (43) in section 7.2).

To demonstrate the dynamics of a realistic RNA molecule we decided to investigate the refolding from local minimum 80 to local minimum 56 (highlighted in gray at the very left of figure 34). The reason for choosing exactly these two states is twofold: First, both have a high energy barrier (7.88 kcal/mol for 80, 8.81 kcal/mol for 56, respectively) and second, they are located in a different subtree than the remaining low-lying local minima in the barrier tree. In order to reach 56 from 80, the molecule must overcome this 7.88 kcal/mol barrier. The interesting feature of this transition is the fact that once the molecule has overcome the first barrier, there are only another 0.93 kcal/mol it has to "climb up" energetically to access the large subtree on the right. Hence, one can expect two very different folding pathways from 80 to 56: A fast one indicating direct transition and a slower one indicating that the molecule crosses the highest saddle point first, then falls down into the right subtree before climbing up again to finally reach local minimum 56. We found exactly this behavior with the coarse-grained approach (figure 35). Population of 80 decreases rapidly starting at 100 time steps and is equally zero after 5×10^4 time steps. Population of basin 56 begins at very early stages of the refolding process (thick curve in figure 35), the initial phase is characterized by a rapid rise of population probability - up to approximately 50 % at 2×10^4 time units. At this point, the absorbing curve gets significantly smoother and forms a pronounced plateau at approximately 53 % within the long time frame from 2×10^4 up to approximately 6×10^5 time steps. This clearly illustrates a second, slower refolding pathway via the right subtree in figure 34. Finally, after 2×10^8 time steps, the refolding process is complete. Note that other local minima are not populated at rates of more than 15 % during the refolding process. We show the population curves of the ground state and local minimum 5 in figure 35. Figure 36 shows a cumulative distribution of first passage times from a Monte Carlo run (average over 9000 `kinfold` simulations) of the refolding from 80 to 56. As in the macrostate approach, a pronounced plateau illustrates the two different folding pathways. It seems fair to say that the macrostate approach is in reasonable agreement with the exact simulation. The time scale of the macro-state process is shifted somewhat to shorter times and the percentage of trajectories that fold directly is overestimated. This is probably a consequence of the truncation of the energy landscape to 1000 states which leads to incomplete sampling of high energy structures that are more likely to lead outside the 56-80 subtree. For transitions with lower energy barriers the agreement is generally better.

Nevertheless we can conclude that the coarse-grained approach has one major

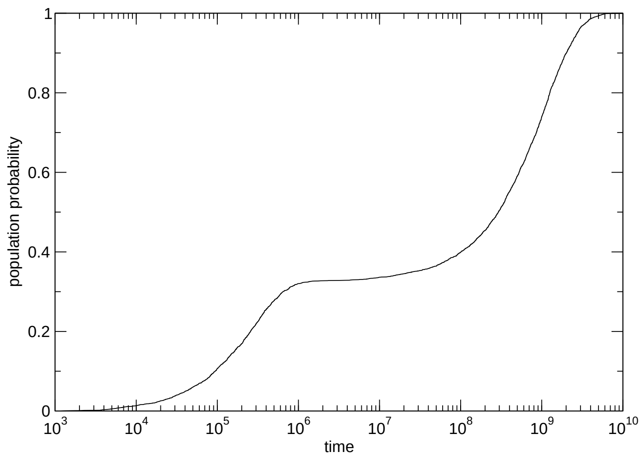


Figure 36: Refolding of a tRNA molecule: Monte Carlo (`kinfold`) simulation of the refolding event from local minimum 80 to local minimum 56. The line shows a cumulative distribution of first passage times to reach 56. A pronounced plateau indicated two different folding pathways.

advantage compared to the exact simulation, namely computational time requirements. The `kinfold` simulations for Fig. 36 required about 3 months of CPU time on an Intel Pentium 4 running at 2.4 GHz under Linux. In the coarse-grained model, the computational bottleneck concerning CPU and memory resources is the diagonalization of the transition matrix $\mathbf{R}$, necessary for the computation of $\exp(t\mathbf{R})$. For 1000 states diagonalization takes on the order of 1 minute.

7.3.2 Lattice Protein Dynamics

After illustrating the capabilities of the coarse-grained dynamics for RNA in the last section, we will focus on lattice proteins in the following and demonstrate that the algorithm is readily applicable to lattice protein folding. We will give representative examples of degenerate energy landscapes of small lattice proteins and compare results from the coarse-grained approach to exact dynamics as calculated with the Monte Carlo algorithm of `pinfold`.

As a first example, we will show the energy landscape and dynamics of a tiny lattice protein ($n = 10$) with sequence `HPPHNXHXPN` (labeled `v01`) on the `SQ` lattice. The conformation space[16] X consists of 2034 structures, that is the number of SAWs of length 10 on the `SQ` lattice after eliminating all structures that are subject to rotation and translation. As one would expect, the associated energy

[16] We chose pivot moves as move set for this example.

landscape is fairly simple, as illustrated in the barrier tree in figure 37. However, even this simple example exhibits salient features of lattice protein barrier trees, namely a large degree of degeneracy (we mentioned this earlier in section 4.1). There are two ground states with an energy of -9 (arbitrary energy units). Local

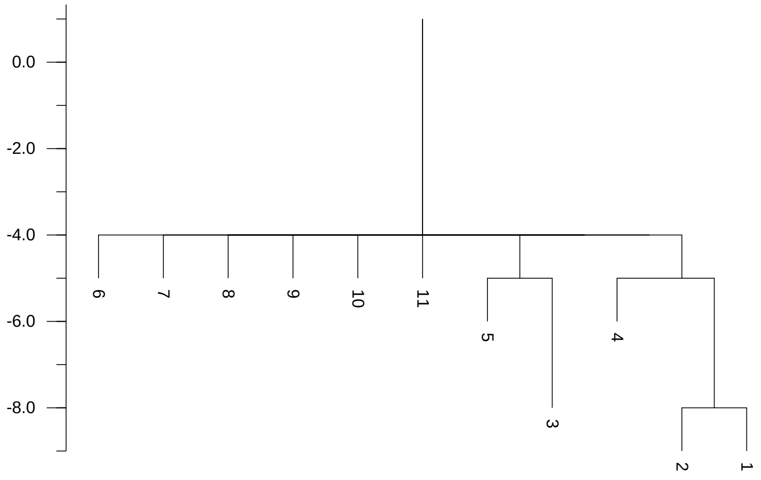


Figure 37: Barrier tree of a short lattice protein on the 2D square lattice (**SQ).**

minimum 4 is directly connected to the two ground states. Local minimum 5 and its father 3 form another "sub-tree", whereas the remaining minima 6-11 are degenerate at an energy level of -5.

We show here the refolding dynamics from 5 to 2, calculated with the Arrhenius and macrostate approach (figure 38). A major difference among tree and rates process is definitely the time scale. The tree process takes place at a region between 10^{-2} and 5×10^4 time units, the macrostate process is shifted towards shorter times at 2 orders of magnitude. Note that this behavior is in contrast to the RNA case where Arrhenius approximations generally tend towards shorter time scales. Within the tree process, basin 5 rapidly loses its fractional population, whereas basins 1, 3 and 4 are populated within the first 100 time steps at values of 21, 6 and almost 52 percent, respectively. Interestingly, the curve of absorbing state 2 shows a distinct shoulder in the region between 10^2 and 10^3 time steps, indicating two folding pathways: A direct one and an indirect one via basin 3. The indirect folding pathway is also supported by the broad shape of the curve of basin 3 which means that - within the tree approach - it is not so easy to escape from basin 3 in order to reach 2. Finally, 3 is de-populated and absorbing state 2 is populated at 100 percent after approximately 5×10^4 time

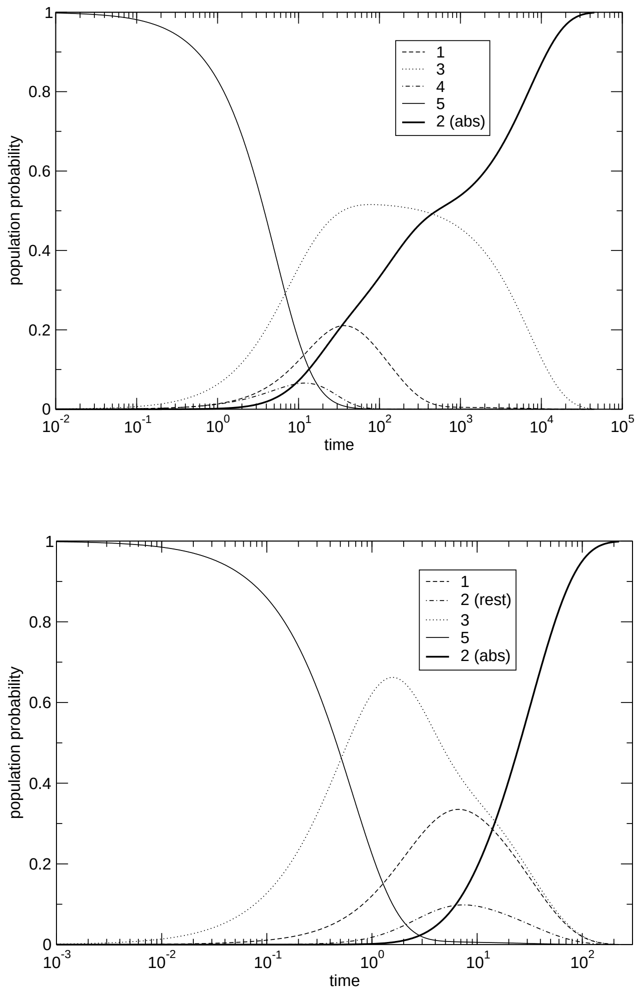


Figure 38: Refolding of a simple lattice protein on the `SQ` lattice. Arrhenius process (upper plot) and macrostate process (lower plot). Local minimum 3 as well as the the 'non-absorbing' states from 2 are populated noticeably during the refolding from 5 to 2.

steps. The rates approach shows quite different dynamics. As expected, the first region is dominated by a rapid loss of fractional population of basin 5 in favor of its direct father, basin 3. Although the energy barrier between 5 and 2 has a height of only 2 (arbitrary) units, basin 3 is populated up to 66 percent after a (very short) time period of 1.6 time units. Nevertheless, basins 1 as well as the non-absorbing states of basin 2 are also populated to a maximum of 33 and 10 percent, respectively at intermediate times. Population of state 2 starts early (1 time unit) and is already finished at 220 time units.

We show the results of a cumulative distribution of first passage times of the refolding from 5 to 2 from a Monte Carlo run (average over 10000 `pinfold` simulations) in figure 39. This approach does not show a pronounced plateau in the folding trajectory, which means that the refolding event happens directly from 5 to 2 without involvement of state 3. Although the distribution of first passage times rather resembles the tree than the macrostate approach, it seems fair to argue that the Monte Carlo simulation lies somewhere in between the coarse-grained approaches presented above. Neither tree nor rates process can approximate the refolding event exactly.

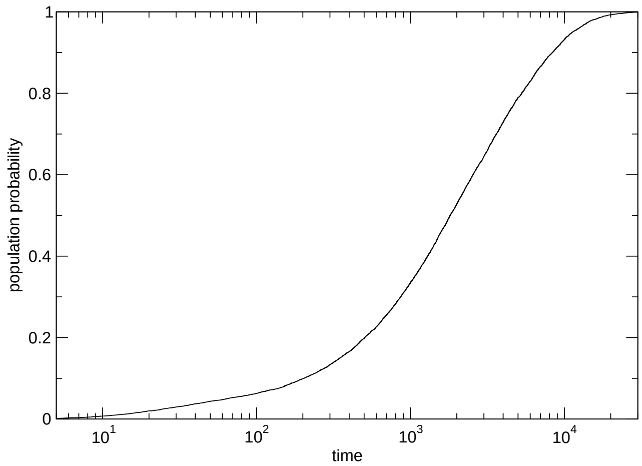


Figure 39: Refolding of the lattice protein `v01`: Monte Carlo (`pinfold`) simulation of the refolding event from local minimum 5 to local minimum 2. The line shows a cumulative distribution of first passage times to reach 2.

The complete refolding path from 5 to 2 is given in figure 40. Similar to figure 33, the upper part illustrates the energy profile, state 5 is located on the very left, 2 on the right. The lower part shows associated SAWs in relative move notation as well as corresponding structures. The first step leads to basin 3 via saddle point

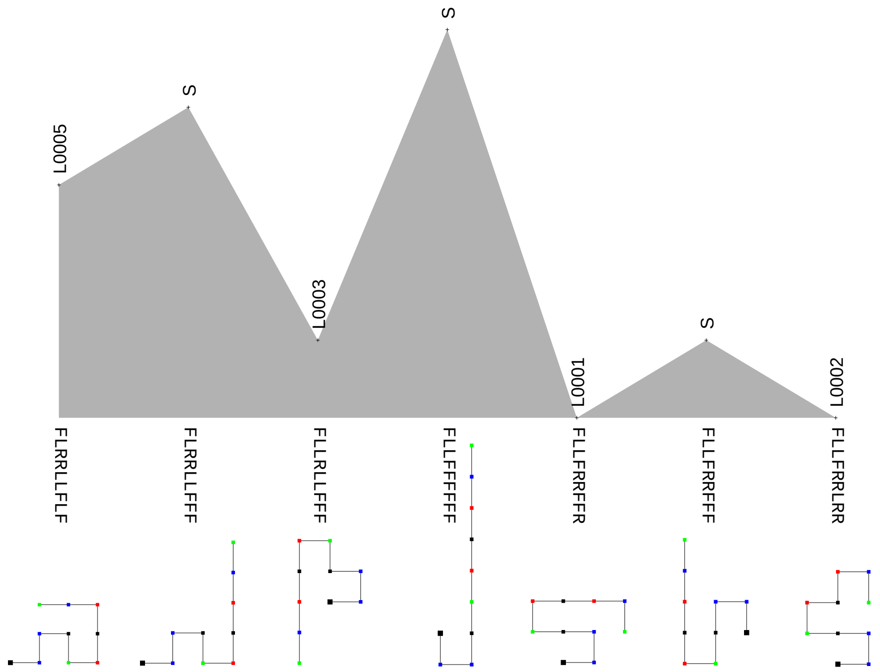


Figure 40: Energy profile of the refolding path from basin 5 to 2. Letter codes in the upper part are analogous to those in figure 33. Relative moves of the SAWs and associated structures are given below. Color code: **H** black, **P** blue, **N** green, **X** red. Basin 1 must be visited in order to reach 2 from 5.

structure `FLRRLLFFF`. After that, it is interesting that the refolding process visits basin 1 prior to reaching 2. The highest energy saddle that needs to be crossed within this refolding event is `FLLFFFFFF` connecting 3 with 1.

The following set of examples is slightly different from the one given before. Instead of demonstrating lattice protein folding for each lattice we implemented we choose one specific sequence from the **HPNX** alphabet and try to determine its dynamic behavior using of different lattices. Although this sounds easy, the choice of such a sequence is highly challenging. On the one hand we are constrained to a short sequence length (since it is computationally impossible (at present) to consider all SAWs of length 27 in order to explore the full energy landscape of the often referenced 27-mer on the **SC** lattice), on the other hand we want to choose a sequence that yields not more than 5 ground states on any lattice we will observe. We decided to choose a sequence with $n = 16$, i.e. `NNHHPPNNPHHHHPXP`, which we labeled `kh68`. Although this sequence is fairly short, it turned out to be appropriate for our requirements. The table below lists the sizes of conformation spaces of the lattices we used to model refolding kinetics.

lattice	total size of conformation space
SQ	802075
HEX	4982
TRI	963627597
TET	3079826

The following figures 41- 44 show the refolding dynamics from selected minima of the barrier tree. For each example, we give the barrier tree showing the 50 lowest-energy states of the associated energy landscape. The middle plot on each page displays the coarse-grained dynamics as approximated with the Arrhenius process, the lower plots show the same transitions assuming macrostate dynamics. Additionally we give the cumulative distribution of first passage times of each transition as calculated by `pinfold` in the lowest plot (red curve). Investigation of the plots yields the following results:

- The general shape and topology of the energy landscape (and hence the barrier tree) is *strongly* lattice-dependent. Sequences that fold into a unique ground state on one lattice may have several degenerate ground states on another lattice.
- The coarse grained dynamics strongly depends on the chosen transition criteria and this dependency is much stronger than for RNA. This fact can easily be read off from figures 41- 44. We can state that not only the population densities of interim populated states is different in the tree and rates process, but also the overall refolding time.
- Direct comparison of the target structures' trajectories with first passage times from `pinfold` (lower plots) give rise to the assumption that the coarse-grained macrostate approach is not *as* suitable in modeling the kinetics as it is for RNA. Note that we assume thermodynamic equilibrium in each basin within our model. Since energy landscapes of lattice proteins are extremely degenerate it seems fair to say it is hard to reach an equilibrium state within a flat landscape. We conjecture that this is the reason for discrepancies detected here. Further investigation is thus necessary.

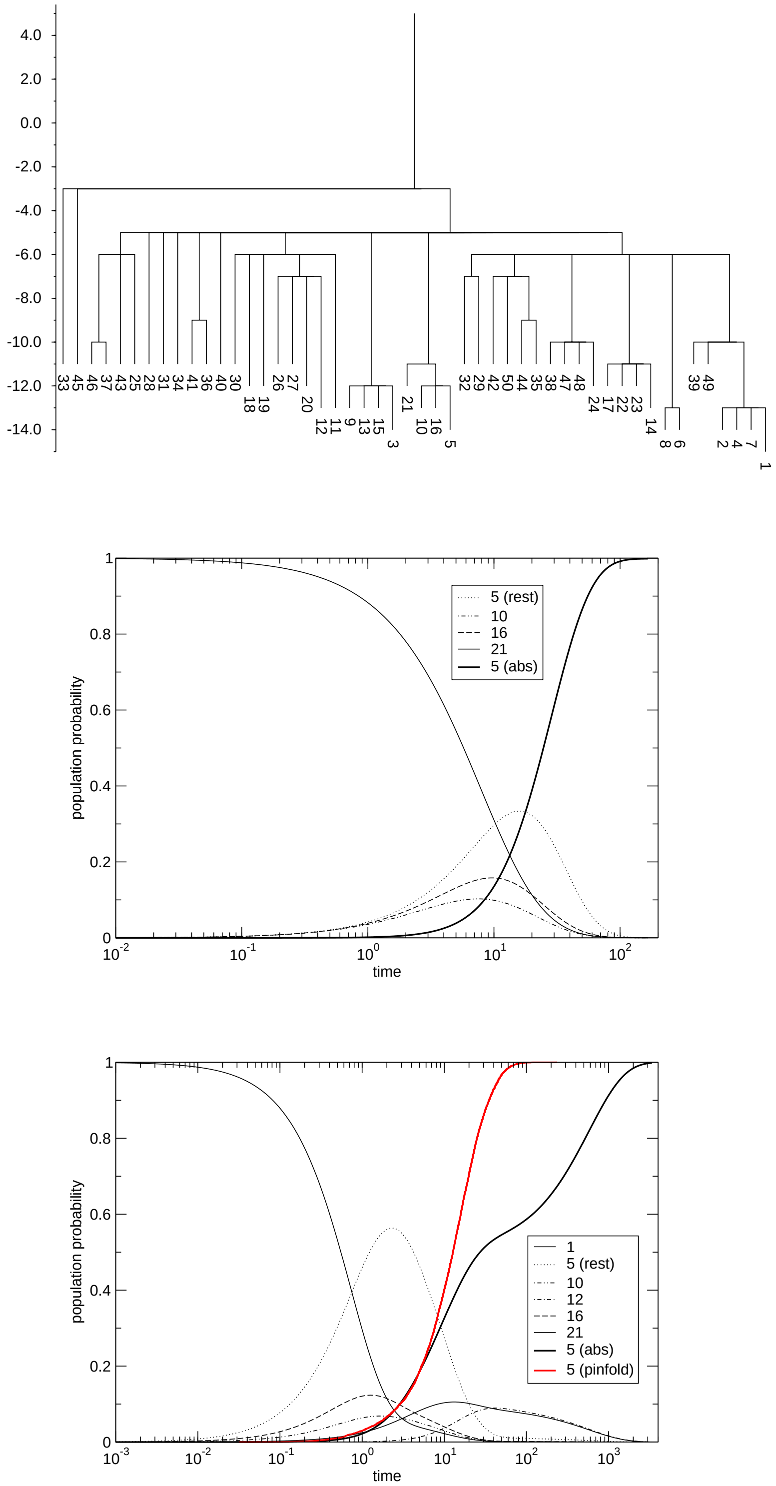


Figure 41: Barrier tree and refolding dynamics of `kh68` with the **SQ** lattice. We show the refolding event from basin 21 to basin 5. Middle plot: Tree process. Lower plot: Macrostate process and Monte Carlo distribution of first passage times of basin 5 (red trajectory).

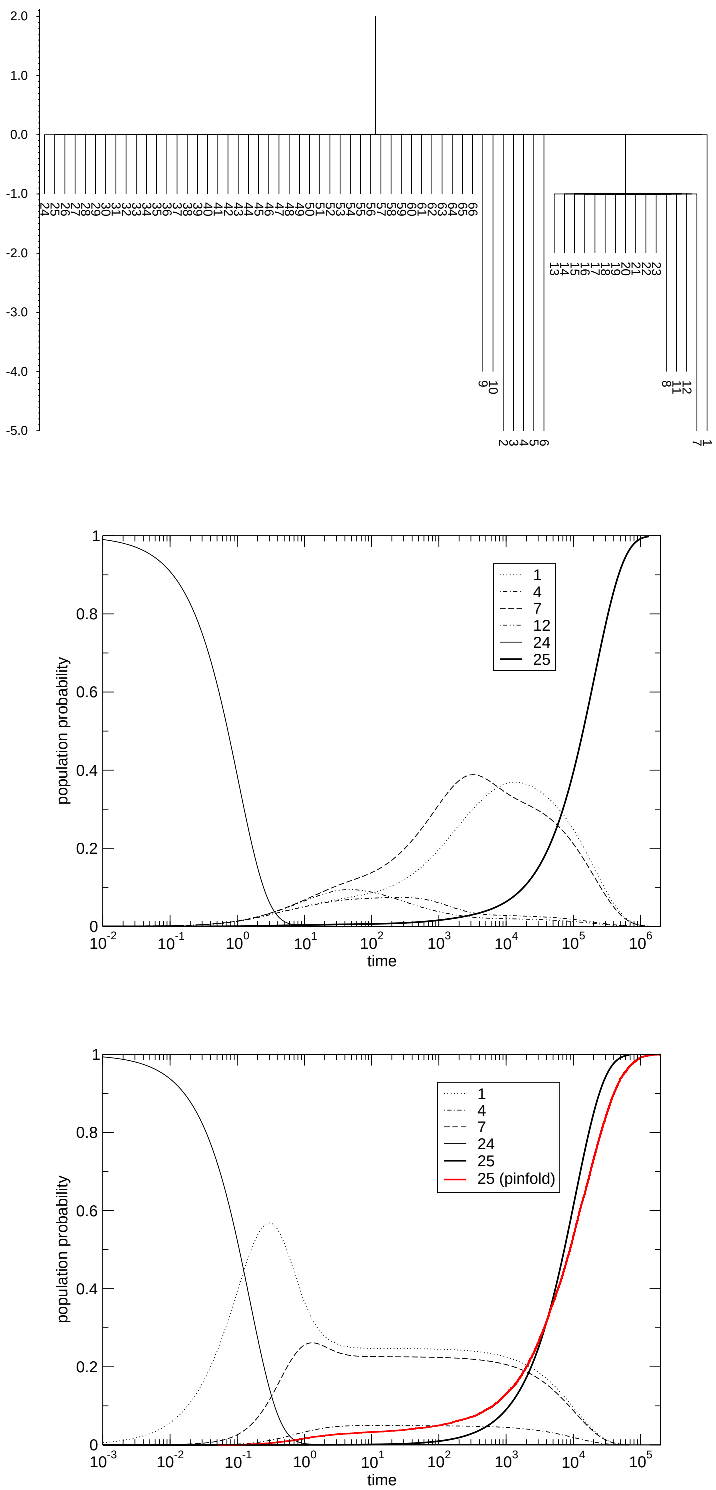


Figure 42: Barrier tree and refolding dynamics of `kh68` with the **HEX** lattice. We show the refolding event from basin 24 to basin 25 (the two leftmost minima in the barrier tree above). Middle plot: Tree process. Lower plot: Macrostate process and Monte Carlo distribution of first passage times of basin 25 (red trajectory).

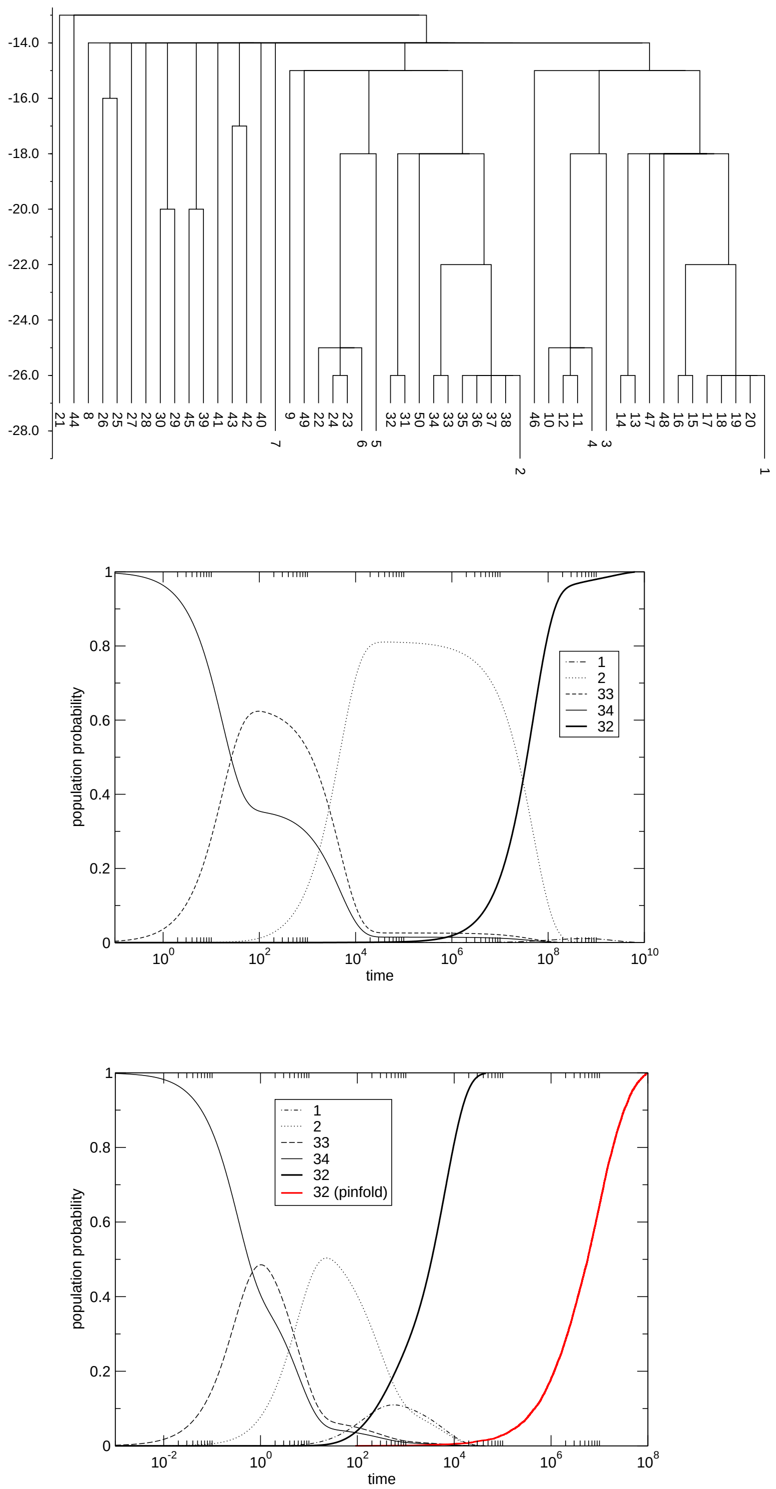


Figure 43: Barrier tree and refolding dynamics of kh68 with the **TRI** lattice. We show the refolding event from basin 34 to basin 32. Middle plot: Tree process. Lower plot: Macrostate process and Monte Carlo distribution of first passage times of basin 32 (red trajectory).

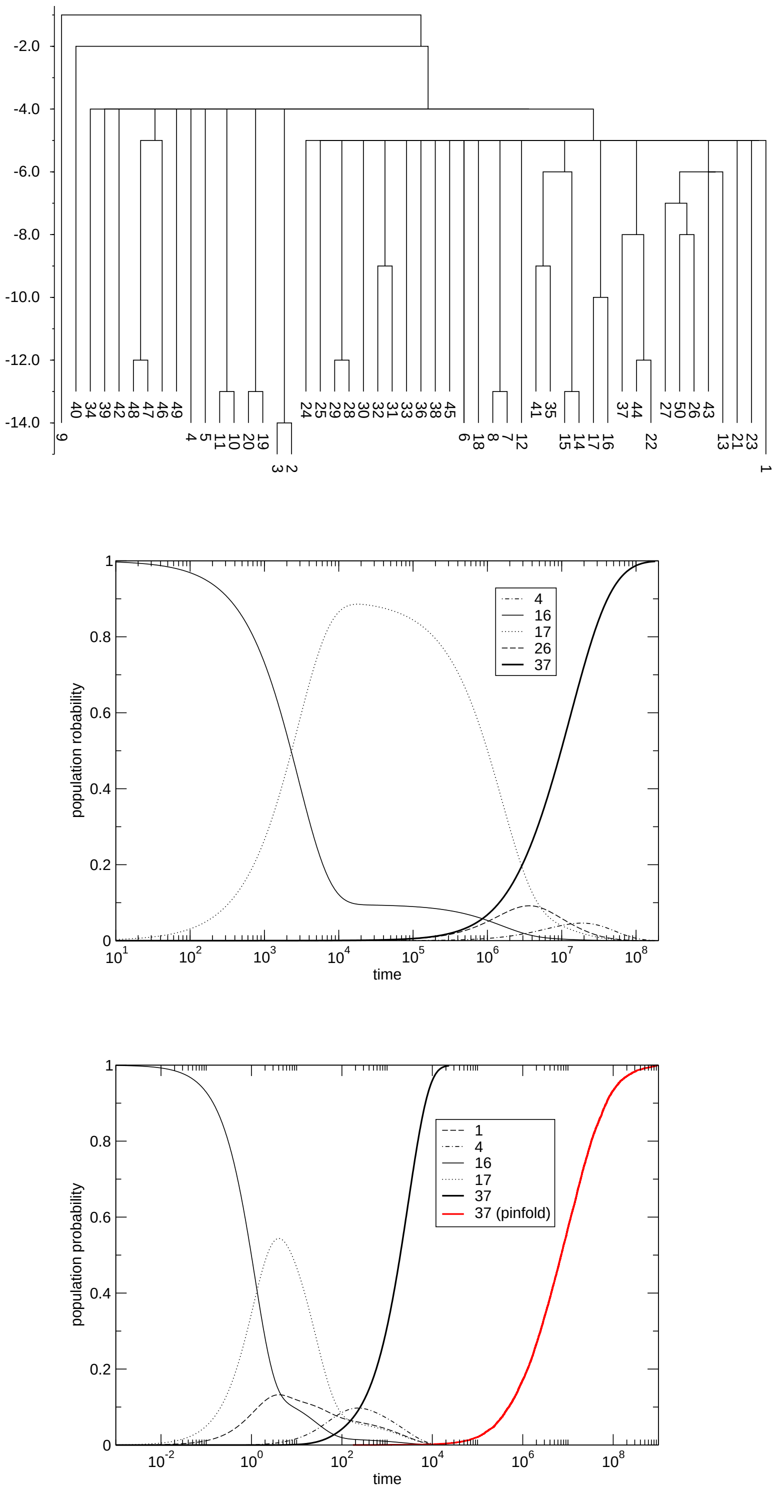


Figure 44: Barrier tree and refolding dynamics of `kh68` with the **TET** lattice. We show the refolding event from basin 16 to basin 37. Middle plot: Tree process. Lower plot: Macrostate process and Monte Carlo distribution of first passage times of basin 37 (red trajectory).

8 Summary and Discussion

8.1 Summary of results

Biopolymers are necessary for the existence of all known forms of life. Amino acids are building blocks for proteins, nucleotides form nucleic acids like DNA or RNA. The ability of biopolymers to fold into a well-defined native state is a prerequisite for biologically functional molecules. RNA secondary structures provide a convenient form of coarse graining, hence their study yields information useful in the prediction of the full three dimensional structure as well as in the interpretation of the biochemical function of the molecules. A simple, yet exact lattice model for proteins is the **HP** model. It distinguishes between hydrophobic (**H**) and polar (**P**) residues and searches for a conformation with a maximal packing of the hydrophobic amino acids.

A fundamental prerequisite in complexity studies of molecular systems is certainly a thorough investigation of the energy surface on which the system dynamics evolve. A detailed understanding of structural features of complex landscapes thus lies at the heart of the biophysics of heteropolymers. Kinetics and structure formation processes of biopolymers are crucially determined by the topological details of the energy landscape, i.e. basins and barriers separating them. The topology of an energy landscape is in turn dependent on a metric used to interconvert structures into each other, called *move set*. The most elementary move set at the level of RNA secondary structures consists of removal and insertion of a single base pair, a slightly more sophisticated move set enables additional base pair shifts. For lattice proteins, the simplest move is achieved by selection of a bead of the lattice chain (pivot point) and rotation of the remaining elements of the chain around a certain angle. We call this inter-conversion pivot move. Crankshaft-, corner- and end moves provide a different move set that has often been used in literature.

A detailed analysis of RNA folding landscapes has become possible by means of an algorithm that generates all RNA secondary structures within a certain energy interval above the ground state. Investigation of lattice protein folding landscapes is constrained to the fact that lattice protein folding is NP-complete. This means that there is no efficient algorithm available to calculate the ground state or a list of suboptimal lattice protein structures above the ground state. To overcome the problem of exhaustive enumeration of structures (which is evidently constrained to very short chain lengths), we developed a tool, `latticeFlooder`, that generates the lower part of the energy landscapes by means of application of elementary moves starting from a low-energy state.

A computer program, `barriers`, to efficiently measure features of energy landscapes (from an energy-sorted list of conformations) such as the number of local minima, the size distribution of basins of attraction or thermodynamic quantities has been developed in our group within the last years. This tool is capable of constructing a hierarchical order of conformations that can be represented compactly in so called *barrier trees*. A barrier tree gives an impression on the shape and ruggedness of the energy landscape and hence shows the distribution and energy ratios of local minima. We described the algorithm of `barriers` and gave a formal definition of degenerate energy landscapes.

Based on elementary steps in conformation space, a stochastic algorithm for the simulation of kinetic folding of RNA has been extended to handle lattice proteins as well (`kinfold/pinfold`). Having this tool for an exact Monte Carlo simulation of folding kinetics of biopolymers at hand, an extended Arrhenius-type kinetics can be formulated on the barrier tree. More precisely, this model allows us to formulate a continuous time Markov process describing population probabilities of different macro states, i.e. local minima of the barrier tree. A major advantage of the so calculated coarse-grained dynamics compared to the exact stochastic method is time efficiency. Investigations of refolding paths that last weeks or months within the exact approach are feasible at a timescale of several minutes within the macrostate approach. This two step strategy consisting of (i) the construction of a barrier tree and (ii) modeling the reaction dynamics on the tree can be carried over to any kind to discrete landscape.

8.2 Discussion and Outlook

At least for the RNA case, the marcostate (barrier tree) conformational kinetics compared favorably with the results of the stochastic simulations. Deviations can be interpreted by inspection of the details of the barrier tree, in particular through computation of the influence of multiple paths between metastable conformations. The situation is different with lattice proteins. Evidently, the simplifications we introduced (i.e. fixed bond lengths and bond angles, drastic reduction of the alphabet size) account not only for a dramatic bias towards degenerate energy landscapes, but also influence the principal ability to model biopolymer folding reasonably. These problems could be overcome by introducing a larger alphabet, preferably with a size of around 10 different types of monomers. However, the problem with bigger alphabets is the choice of a proper interaction potential. Another possibility to circumvent degeneracies in lattice protein energy landscapes would be a choice of more realistic lattices, i.e. those with higher coordination numbers. The drawback with this approach would be, however, a dramatic increase of computational requirements (remember that we need a list of all structures within an energy interval above the ground state in order to execute the algorithm from section 4.2). Studying lattice protein dynamics generally showed different behavior for the coarse-grained (Arrhenius vs. macrostate) dynamics . Although some results are in reasonable agreement with results from our Monte Carlo dynamics simulations, we cannot derive a general rule what combinations of lattices, move sets or coarse grained approaches are more appropriate for a given sequence. On the other side, we can conclude that the general shape and topology of an energy landscape is strongly lattice-dependent. In other words it is impossible to deduce kinetic folding properties of a certain lattice protein sequence without a thorough investigation of energy landscapes with different lattices. Note that this fact even applies to small lattice protein sequences, as those that were used here. It is consequently fair to argue that model studies for very distinct protein families that have been reported in literature (e.g. [29, 63, 142]) might exhibit completely different folding behavior when applied to different lattices. In other words we postulate that it is necessary to choose certain lattices for different kinds of protein models.

A challenging aspect in protein folding is definitely the partitioning between "good" and "bad" folders [149]. With good folders we mean sequences that converge systematically towards a unique native conformation and do so within reasonable time. Having lattice protein dynamics at hand, it would be interesting to derive features that are specific for good/bad folders from the (macrostate) dynamics. Within this framework, another fundamental aspect would be to find out if there is correlation between the energy landscape on the one side and the dynamics on the other, i.e. can one derive general rules for the dynamic behavior with knowledge of the energy landscape's topology. As a first approach one would try to calculate properties like the ratio T_f/T_g from the energy function.

We mentioned earlier that the concept of folding funnels needs further refinement (chapter 5). Several approaches are within reach to address this problem. We could, for example, extend the level of coarse-graining on the barrier tree. Remember section 7.1 where we chose gradient basins as macrostates. We could of course constrain our macrostates to local minima of the barrier tree that have a (predefined) minimum barrier height, leading from our original partition $\mathbf{\Pi} = \{\alpha, \beta, \gamma, \ldots\}$ to a modified partition $\mathbf{\Pi}' = \{\alpha', \beta', \gamma', \ldots\}$. Increasing the minimum barrier height reduces the number of macrostates and consequently increases the entropy of each macrostate. Plotting minimum barrier height against conformational entropy of the (still available) macrostates should thus yield a step-function which, mirrored at the abscissa and rotated 90 degrees counterclockwise, should yield a modified "folding funnel with well-defined lateral expansion". However, this is just one possible approach towards a mathematical/thermodynamic foundation of folding funnels.

Appendix A

Energy matrices for different alphabets:

HP

	H	P
H	-1	0
P	0	0

HP′[1]

	H	P
H	-3	-1
P	-1	0

HPNX[2]

	H	P	N	X
H	-4	0	0	0
P	0	1	-1	0
N	0	-1	1	0
X	0	0	0	0

YhHX[1]

	h	H	Y	X
h	-2	-4	-1	2
H	-4	-3	-1	0
Y	-1	-1	0	2
X	2	0	2	0

[1] taken from ref. [10]

[2] taken from ref. [3] due to compliance and direct comparison to Backofen's method

Appendix B

A comprehensive list of energy terms generally used in force fields:

Bond - Energy:

The energy between two bonded atoms increases, when the bond is compressed or stretched. The potential is described by an equation based on Hooke's law for springs.

$$E_{bond} = \sum_{bonds} k_b(r - r_0)^2$$

whereby k_b is the force constant, r is the actual bond length and r_0 the equilibrium length. This quadratic approximation fails as the bond is stretched towards the point of dissociation.

Angle Energy:

Energy increases if the equilibrium bond angles are bent. Again the approximation is harmonic and uses Hooke's law.

$$E_{angle} = \sum_{angles} k_\theta(\theta - \theta_0)^2$$

k_θ controls the stiffness of the angle, θ is the current bond angle and θ_0 the equilibrium angle. Both, the force and equilibrium constant have to be estimated for each triple of atoms.

Torsion Energy:

Intra-molecular rotations (around torsions or dihedrals) require energy as well:

$$E_{torsion} = \sum_{torsions} \frac{V_n}{2}(1 + cos(n\omega - \gamma))$$

V_n controls the amplitude of this periodic function, n is the multiplicity, and γ the so-called phase factor, shifts the entire curve along the rotation angle axis ω. Again the parameters V_n, n and γ for all combinations of four atoms have to be determined.

Non-bonding Energy:

The simplest potential for non-bonding interactions includes two terms, a Van der Waals and a Coulomb term.

$$E_{non-bonding} = \underbrace{\sum_i \sum_{j>i} \left(\frac{A_{ij}}{r_{ij}^6} - \frac{B_{ij}}{r_{ij}^{12}} \right)}_{\text{Van der Waals}} + \underbrace{\sum_i \sum_{j>i} \frac{q_i q_j}{r_{ij}}}_{Coulomb}$$

The Van der Waals term accounts for the attraction and the Coulomb term for electrostatic interaction. The shown approximation for the van der Waals energy is of the Lennard-Jones 6-12 potential type.

Appendix C

The *refolding* path from local minimum 46 to the ground state of the RNA switch lz04 from section 7.3.1. S denote saddle points, I intermediate structures and L local minima.

```
.....((.(((((((...(....)...))))))).))....(((((....))))).... (-15.20) L0046
......(.(((((((...(....)...))))))).).....(((((....))))).... (-14.50) S
........(((((((...(....)...)))))))).......(((((....))))).... (-15.60) L0040
........(((((((............)))))))).......(((((....))))).... (-15.20) I
........((((((......(......)))))))).......(((((....))))).... ( -9.80) S
........(((((......((......)))))))).......(((((....))))).... (-10.00) L0322
........((((......(((......)))))))).......(((((....))))).... ( -9.30) S
........(((......((((......)))))))).......(((((....))))).... ( -9.60) I
........((......(((((......)))))))).......(((((....))))).... (-10.70) L0242
........(......((((((......)))))))).......(((((....))))).... (-10.00) S
........(......((((((......))))))........(((((....)))))...) (-12.00) I
...............((((((......))))))........(((((....))))).... (-14.80) I
..............(((((((......)))))))).......(((((....))))).... (-15.60) I
.............((((((((......))))))))......(((((....))))).... (-19.90) L0002
..............(((((((......)))))))).......(((((....))))).... (-15.60) I
...............((((((......))))))........(((((....))))).... (-14.80) I
................(((((......))))).........(((((....))))).... (-13.40) I
.................((((......))))..........(((((....))))).... (-12.40) I
.................(((........)))..........(((((....))))).... (-10.70) I
.................((..........))..........(((((....))))).... ( -9.20) I
.................(............)..........(((((....))))).... ( -6.80) S
..............................(......)...(((((....))))).... ( -7.20) I
.........................................(((((....))))).... ( -9.50) L0400
..............................(......)...(((((....))))).... ( -7.20) S
..............................((....))...(((((....))))).... ( -7.60) I
.............................(((....)))..(((((....))))).... ( -8.90) I
............................((((....)))).(((((....))))).... (-10.10) L0305
............(...............((((....)))).(((((....)))))...) ( -7.50) I
............(........)......((((....)))).(((((....))))).... ( -7.00) S
............((......))......((((....)))).(((((....))))).... ( -7.80) I
............(((....)))......((((....)))).(((((....))))).... ( -9.00) L0490
...........((((....))).)....((((....)))).(((((....))))).... ( -8.40) I
.........(.((((....))).).)..((((....)))).(((((....))))).... ( -6.50) S
........((.((((....))).).)).((((....)))).(((((....))))).... ( -9.30) I
.......(((.((((....))).).)))((((....)))).(((((....))))).... (-10.80) L0232
.......(((.((((....))).).))).(((....)))..(((((....))))).... ( -9.60) I
.......(((.((((....))).).)))..((....))...(((((....))))).... ( -8.30) I
.......(((.((((....))).).)))..(......)...(((((....))))).... ( -7.90) S
...(...(((.((((....))).).)))..)..........(((((....))))).... ( -9.40) I
.......(((.((((....))).).))).............(((((....))))).... (-10.20) L0299
...(...(((.((((....))).).)))..)..........(((((....))))).... ( -9.40) I
...(...(((.((((....))).).))).)...........(((((....))))).... ( -8.60) S
...((..(((.((((....))).).)))))...........(((((....))))).... ( -9.60) I
..(((..(((.((((....))).).))))))..........(((((....))))).... (-10.10) I
.((((..(((.((((....))).).))))))).........(((((....))))).... (-11.00) L0223
.(((((..((.((((....))).).))))))).........(((((....))))).... (-10.20) I
.((((((..(.((((....))).).))))))).........(((((....))))).... ( -8.90) S
.((((((....((((....))).)..)))))).........(((((....))))).... (-11.90) L0158
.((((((.....(((....)))....)))))).........(((((....))))).... (-11.40) I
.((((((.....((......))....)))))).........(((((....))))).... (-10.20) I
.((((((.....(........)....)))))).........(((((....))))).... ( -9.40) S
.((((((....(.........)....)))))).........(((((....))))).... (-10.00) I
.((((((...................)))))).........(((((....))))).... (-10.41) L0271
.((((((.....(........)....)))))).........(((((....))))).... ( -9.40) I
.((((((.(............)....)))))).........(((((....))))).... ( -8.60) S
.((((((.((..........))....)))))).........(((((....))))).... (-10.20) I
.(((((((((..........)))...)))))).........(((((....))))).... (-11.70) L0170
.(((((((((..........))))...))))).........(((((....))))).... (-11.00) S
.(((((((((..........)))))...)))).........(((((....))))).... (-11.20) L0205
.(((((((((..........))))))...))).........(((((....))))).... (-10.70) S
.(((((((((..........)))))))...)).........(((((....))))).... (-11.30) I
.(((((((((..........))))))))...).........(((((....))))).... (-11.80) I
.(((((((((..........))))))))).............(((((....))))).... (-17.70) I
((((((((((..........))))))))))...........(((((....))))).... (-20.60) L0001
```

List of Figures

List of Algorithms

References

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structures with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] R. Backofen. A Polynomial Time Upper Bound for the Number of Contacts in the HP-Model on the Face-Centered-Cubic Lattice (FCC). *Journal of Discrete Algorithms*, 2003.

[3] R. Backofen and S. Will. A Branch-and-Bound Constraint Optimization Approach to the HPNX Structure Prediction Problem. Technical Report 9810, Ludwig-Maximilians-Universität München, Institut für Informatik, 1998.

[4] R. Backofen and S. Will. A Constraint-Based Approach to Structure Prediction for Simplified Protein Models that Outperforms Other Existing Methods. In *Proceedings of the 19th International Conference on Logic Programming (ICLP 2003)*, 2003.

[5] R. Backofen, S. Will, and P. Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing (PSB 2000)*, volume 5, pages 92–103, 2000.

[6] T. Baumstark, A. R. Schroder, and D. Riesner. Viroid processing: Switch from cleavage to ligation is driven by a change from a tetraloop to a loop E conformation. *EMBO J.*, 16:599–610, 1997.

[7] O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *J. Chem. Phys.*, 106:1495–1517, 1997.

[8] A. Beretti and A. D. Sokal. New Monte Carlo Method for the Self-Avoiding Walk. *J. Stat. Phys.*, 40:483–531, 1985.

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

[10] E. Bornberg-Bauer. Chain growth algorithms for HP-type lattice proteins. In *First Annual International Conference on Computational Molecular Biology (RECOMB), Santa Fe, NM, USA*, pages 47–55, NY/USA (1997), 1997. ACM Press.

[11] J. U. Bowie, R. Lüthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–169, 1991.

[12] R. R. Breaker. Engineered allosteric ribozymes as biosensor components. *Curr. Opin. Biotechnol.*, 13:31–39, 2002.

[13] N. Breton, C. Jacob, and P. Daegelen. Prediction of sequentially optimal RNA secundary structures. *J. Biomol. Struct. Dyn.*, 14:727–740, 1997.

[14] P. Brion and E. Westhof. Hierarchy and dynamics of RNA folding. *Annu. Rev. Biophys. Biomol. Struct.*, 26:113–137, 1997.

[15] B. R. Brooks, R. E. Bruccoleri, B. D Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy minimization and dynamics calculations. *J. Comp. Chem.*, 4:187–217, 1983.

[16] J. D. Bryngelson and P. G. Wolynes. Spin glasses and the statistical mechanics of protein folding. *Proc. Natl. Acad. Sci. USA*, 84:7524–7528, 1987.

[17] P. Butera and M. Comi. N-vector spin models on the simple-cubic and body-centered-cubic lattices: A study of the critical behavior of susceptibility and of the correlation length by high-temperature series extended to order β^{21}. *Phys. Rev. B*, 56:8212–8240, 1997.

[18] C. J. Camacho and D. Thirumalai. Kinetics and thermodynamics of folding in model proteins. *Proc. Natl. Acad. Sci. USA*, 90:6369–6372, 1993.

[19] E. R. Canfield. Remarks on an asymptotic method in combinatorics. *J. Combin. Theory A*, 37:348–352, 1984.

[20] N. Carmi, R. Balkhi, and R. R. Breaker. Cleaving DNA with DNA. *Proc. Natl. Acad. Sci.*, 95:2233–2237, 1998.

[21] O. Catoni. Rough large deviation estimates for simulated annealing: Application to exponential schedules. *Ann. Probab.*, 20:1109–1146, 1992.

[22] O. Catoni. Simulated annealing algorithms and Markov chains with rate transitions. In J. Azema, M. Emery, M. Ledoux, and M Yor, editors, *Seminaire de Probabilites XXXIII*, volume 709 of *Lecture Notes in Mathematics*, pages 69–119. Springer, Berlin/Heidelberg, 1999.

[23] T. Cech. RNA as an enzyme. *Scientific American*, 11:76–86, 1986.

[24] H. S. Chan and K. A. Dill. Intrachain loops in polymers: Effects of excluded volume. *J. Chem. Phys.*, 90(1):492–509, 1989.

[25] H. S. Chan and K. A. Dill. "sequence space soup" of proteins an copolymers. *J. Chem. Phys*, 95(5):3775–3787, 1991.

[26] H. S. Chan and K. A. Dill. Protein folding in the landscape perspective: Chevron plots and non-Arrhenius kinetics. *Proteins: Structure, Function, and Genetics*, 30:2–33, 1998.

[27] M. Chen and K. Y. Lin. Universal amplitude ratios for three-dimensional self-avoiding walks. *J. Phys. A: Math. Gen.*, 35:1501–1508, 2002.

[28] M. Clark, R. D. Cramer III, and N. van Obdenbusch. Validation of the general purpose Tripos 5.2 force field. *J. Comp. Chem.*, 10:982–1012, 1989.

[29] C. Clementi, A. E. Garcia, and J. N. Onuchic. Interplay among tertiary contacts, secondary structure formation and side-chain packing in the protein folding meschanism: All-atom representation study of protein L. *J. Mol. Biol.*, 326:933–954, 2003.

[30] P. Crescenzi, D. Goldman, C. Papadimitrou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. In *Proceedings of the Second International Conference on Computational Molecular Biology (RECOMB)*, New York, March 1998.

[31] G. M. Crippen. Easily searched protein folding potentials. *J. Mol. Biol.*, 260(3):467–475, 1996.

[32] J. Cupal, I. L. Hofacker, and P. F. Stadler. Dynamic programming algorithm for the density of states of RNA secondary structures. In R. Hofstädt, T. Lengauer, M. Löffler, and D. Schomburg, editors, *Computer Science and Biology 96 (Prooceedings of the German Conference on Bioinformatics)*, pages 184–186, Leipzig, Germany, 1996. Universität Leipzig.

[33] K. A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501–1509, 1995.

[34] K. A. Dill. Polymer principles and protein folding. *Protein Sci.*, 8:1166–1180, 1999.

[35] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, P. D. Yee, P. D. Thomas, and H. S. Chan. Principles of protein folding - a perspective from simple exact models. *Prot. Sci.*, 4:561–602, 1995.

[36] K. A. Dill and H. S. Chan. From levinthal to pathways to funnels. *Nature Struct. Biol.*, 4(1):10–19, 1997.

[37] A. R. Dinner, A. Šali, and M. Karplus. The folding mechanism of larger model proteins: Role of native structure. *Proc. Natl. Acad. Sci. USA*, 93:8356–8361, 1996.

[38] R. M. Dirks and N. A. Pierce. A parition function algorithm for nucleic acid secondary struc ture including pseudoknots. *J. Comput. Chem.*, 24:1664–1677, 2003.

[39] M. Doi. *Introduction to Polymer Physics.* Clarendon Press, Oxford, 1996.

[40] T. Doslic, D. Svrtan, and D. Veljan. Enumerative aspects of secondary structures. *Discrete Mathematics*, 285:67–82, 2004.

[41] J. P. Doye, M. A. Miller, and D. J. Welsh. Evolution of the potential energy surface with size for Lennard-Jones clusters. *J. Chem. Phys.*, 111:8417–8429, 1999.

[42] R. Du, A.Y. Grosberg, T. Tanaka, and M. Rubinstein. Unexpexted Scenario of Glass Transition in Polymer Globules: An Exaxtly Enumerable Model. *Phys. Rev. Lett.*, 84:2417–2420, 2000.

[43] D. M. Dykxhoorn, C. D. Novina, and P. A. Sharp. Killing the messenger: short RNAs that silence gene expression. *Nature Rev. Mol. Cell Biol.*, 4:457–467, 2003.

[44] I. G. Enting and A. J. Guttmann. Self-avoiding polygons on the square, L and Manhatten lattice. *J. Phys. A: Math. Gen.*, 18:1007–1017, 1985.

[45] K. Fan and W. Wang. What is the Minimum Number of Letters Required to Fold a Protein. *J. Mol. Biol.*, 328:921–926, 2003.

[46] G. Fayat, F. J. Mayaux, C. Sacerdot, M. Fromant, M. Springer, M. Grunberg-Manago, and Blanquet S. Escherichia coli phenylalanyl-tRNA synthetase operon region. Evidence for an attenuation mechanism. Identification of the gene for the ribosomal protein L20. *J. Mol. Biol.*, 171:239–261, 1983.

[47] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding kinetics at elementary step resolution. *RNA*, 6:325–338, 2000.

[48] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, 2001.

[49] C. Flamm, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, 216:155–173, 2002.

[50] P. J. Flory. *Statistical Mechanics of Chain Molecules.* Wiley, New York, 1969.

[51] S. M. Freier, R. Kierzek, J. A. Jaeger, N. Sugimoto, M. H. Caruthers, T. Neilson, and D. H. Turner. Improved free-energy parameters for prediction of RNA duplex stability. *Proc. Natl. Acad. Sci. USA*, 83:9373–9377, 1986.

[52] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[53] P. Garstecki, Hoang T. X., and M. Cieplak. Energy landscapes, supergraphs, and “folding funnels” in spin systems. *Phys. Rev. E*, 60:3219–3226, 1999.

[54] W. Gilbert. The RNA world. *Nature*, 319:618, 1986.

[55] N. Gō and H. Taketomi. Respective roles of short- and long-range interactions in protein folding. *Proc. Natl. Acad. Sci. USA*, 75:559–563, 1978.

[56] A. Godzik, A. Kolinski, and J. Skolnick. Lattice representations of globular proteins: How good are they? *J. Comput. Chemistry*, 14:1194–1202, 1993.

[57] G. H. Golub and C. F. Van Loan. *Matrix Computations*, chapter 7, The Unsymmetric Eigenvalue Problem. The John Hopkins University Press, 3 edition, 1996.

[58] S. Govindarajan and R. A. Goldstein. Searching for foldable protein structures using optimized energy functions. *Biopolymers*, 36:43–51, 1995.

[59] S. Govindarajan and R. A. Goldstein. Why are some protein structures so common? *Proc. Natl. Acad. Sci. USA*, 93:3341–3345, 1996.

[60] C. Guerrier-Takada and S. Altman. Catalytic activity of an RNA molecule prepared by transcription in vitro. *Science*, 223:285–286, 1984.

[61] C. Guerrier-Takada, K. Gardiner, T. Marsh, N. Pace, and S. Altman. The RNA moiety of ribunuclease p is the catalytic subunit of the enzyme. *Cell.*, 35:849–857, 1983.

[62] A. P. Gultyaev, F. H. D. van Batenburg, and C. W. A. Pleij. An approximation of loop free energy values of RNA H-pseudoknots. *RNA*, 5:609–617, 1999.

[63] Z. Guo and C. L. Brooks III. Thermodynamics of protein folding: a statistical mechanical study of a small all-β protein. *Biopolymers*, 42:745–57, 1997.

[64] A. J. Guttmann. On the critical behaviour of self-avoiding walks. *J. Phys. A: Math. Gen.*, 20:1839 – 1854, 1986.

[65] A. J. Guttmann. The high-temperature susceptibility end spin-spin correlation function of the three-dimensional Ising model. *J. Phys. A: Math. Gen.*, 20:1855–1863, 1987.

[66] A. J. Guttmann, T. R. Osborn, and A. D. Sokal. Connective constant of the self-avoiding walk on the triangular lattice. *J. Phys. A: Math. Gen.*, 19:2591–2598, 1986.

[67] D. Hamada, S. Segawa, and Goto Y. Non-native alpha-helical intermediate in the refolding of beta-lactoglobulin: A predominantly beta-sheet protein. *Nature Struct. Biol.*, 3:868–873, 1996.

[68] G. J. Hannon. RNA interference. *Nature*, 418:244–251, 2002.

[69] J. P. Hansen and I. R. MacDonald. *Theory of simple liquids.* Academic Press Inc., London, 2nd ed. edition, 1986.

[70] W. E. Hart and S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal. *J. Comput. Biol.*, 3:53–96, 1996.

[71] C. Haslinger. *Prediction algorithms for restricted RNA pseudoknots.* PhD thesis, Universität Wien, 2001.

[72] B. Hayes. Prototeins. *American Scientist*, 86(3):216–221, 1998.

[73] M. Hendlich, P. Lackner, S. Weitckus, H. Floeckner, R. Froschauer, K. Gottsbacher, G. Casari, and M. J. Sippl. Identification of native protein folds amongst a large number of incorrect models — the calculation of low energy conformations from potentials of mean force. *J. Mol. Biol.*, 216:167–180, 1990.

[74] P. G. Higgs. RNA secondary structure: physical and computational aspects. *Quart. Rev. Biophys.*, 33:199–253, 2000.

[75] P. Hobza and Jiří Šponer. Towards true DNA base-stacking energies: MP2, CCSD(T), and complete basis set calculations. *J. Amer. Chem. Soc.*, 124:11802–11808, 2002.

[76] I. L. Hofacker. The Vienna RNA secondary structure server. *Nucl. Acids Res.*, 31:3429–3431, 2003.

[77] I. L. Hofacker, W. Fontana, P. F. Stadler, S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, 125:167–188, 1994.

[78] I. L. Hofacker, W. Fontana, P. F. Stadler, and P. Schuster. `Vienna RNA Package`. `http://www.tbi.univie.ac.at/~ivo/RNA/`, 1994-2004. (Free Software).

[79] I. L. Hofacker, P. Schuster, and P. F. Stadler. Combinatorics of RNA secondary structures. *Discrete Applied Mathematics*, 88:207–237, 1998.

[80] J. A. Howell, T. F. Smith, and M. S. Waterman. Computation of generating functions for biological molecules. *SIAM J. Appl. Math.*, 39:119–133, 1980.

[81] J. A. Jaeger, D. H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. USA*, 86:7706–7710, 1989.

[82] G. F. Joyce. RNA evolution and the origins of life. *Nature*, 338:217–224, 1989.

[83] G. F. Joyce. The rise and fall of the RNA world. *The New Biologist*, 3:399–407, 1991.

[84] K. Kawasaki. Diffusion constants near the critical point for time-dependent Ising models. *Phys. Rev.*, 145:224–230, 1966.

[85] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimation by simulated annealing. *Sciene*, 220:671–680, 1983.

[86] T. Klotz and S. Kobe. Exact low-energy landscape and relaxation phenomena in Ising spin glasses. *Acta Physica Slovaca*, 44:347–356, 1994.

[87] T. Klotz and S. Kobe. "Valley Structures" in the phase space of a finite 3D Ising spin glass with $\pm i$ interactions. *J. Phys. A: Math. Gen*, 27:L95–L100, 1994.

[88] A. Kolinski and J. Skolnick. Reduced models of proteins and their applications. *Polymer*, 45:511–524, 2004.

[89] M. Lal. Monte Carlo computer simulation of chain molecules. *Molec. Phys.*, 17:57–64, 1969.

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

[91] A. L. Lehninger, D. L. Nelson, and M. M. Cox, editors. *Principles of Biochemistry.* Worth Publishers, 3rd edition, 2000.

[92] P. E. Leopold, M. Montal, and J. N. Onuchic. Protein folding funnels: A kinetic approach to the sequence-structure relationship. *Proc. Natl. Acad. Sci. USA*, 89:8721–8725, 1992.

[93] C. Levinthal. Are there pathways for protein folding? *J. Chim. Phys.*, 65:44–45, 1968.

[94] M. Levitt. A simplified represenation of protein conformations for rapid simulation of protein folding. *J. Mol. Biol.*, 104:59–107, 1976.

[95] M. Levitt and A. Warshel. Computer simulation of protein folding. *Nature*, 253:694–698, 1975.

[96] J. Leydold and P. F. Stadler. Minimal cycle basis, outerplanar graphs. *Elec. J. Comb.*, 5:R16, 1998. See `http://www.combinatorics.org`.

[97] H. Li, R. Helling, C. Tang, and N. Wingreen. Emergence of preferred structures in a simple model of protein folding. *Science*, 273:666–669, 1996.

[98] H. Li, C. Tang, and N. S. Wingreen. Designability of Protein Structures: A Lattice-Model Study using the Miyazawa-Jernigan Matrix. *Proteins*, 49:403–412, 2002.

[99] J. S. Lodmell and A. E. Dahlberg. A conformational switch in Escherichia coli 16S ribosomal RNA during decoding of messenger RNA. *Science*, 277:1262–1267, 1997.

[100] R. B. Lyngsø and C. N. S. Pedersen. RNA pseudoknot prediction in energy based models. *J. Comp. Biol.*, 7(3/4):409–428, 2000.

[101] A. D. Mac Kerell Jr., B. Brooks, C. L Brooks III, L. Nilsson, B. Roux, Y. Won, and M. Karplus. CHARMM: The energy function and its parametrization with an overview of the program. In P. v. R. Schleyer, editor, *The Encyclopedia of Computational Chemistry*, volume 1, pages 271–277. John Wiley & Sons: Chichester, 1998.

[102] N. Madras, A. Orlitsky, and L. A. Shepp. Monte Carlo generation of self-avoidng walks with fixed endpoints and fixed length. *J. Stat. Phys.*, 58:159–183, 1990.

[103] N. Madras and A. D. Sokal. Nonergodicity of local, length-conserving Monte Carlo algorithms for the Self-Avoiding Walk. *J. Stat. Phys.*, 47:573–595, 1987.

[104] N. Madras and A. D. Sokal. The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk. *J. Stat. Phys.*, 50:109–189, 1988.

[105] M. Mandal and R. R. Breaker. Gene regulation by riboswitches. *Nat. Rev. Mol. Cell Biol.*, 5:451–463, 2004.

[106] D. H. Mathews, J. Sabina, M. Zuker, and H. Turner. Expanded sequence dependence of thermodynamic parameters provides robust prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, 1999.

[107] S. L. Mayo, B. D. Olafson, and W. A. Goddard III. DREIDING: A generic force field for molecular simulations. *J. Phys. Chem.*, 94:8897–8909, 1990.

[108] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.

[109] A. Meir and J. W. Moon. On an asymptotic method in enumeration. *J. Combin. Theory A*, 51:77–89, 1989.

[110] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing maschines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[111] L. Mirny and E. Shakhnovich. Protein folding theory: From lattice to all-atom models. *Annu. Rev. Biophys. Struct.*, 30:361–96, 2001.

[112] A. E. Mirsky and L. Pauling. On the structure of native, denatured, and coagulated proteins. *Proc. Natl. Acad. Sci.*, 22:439–447, 1936.

[113] S. Miyazawa and R. L. Jernigan. Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation. *Macromolecules*, 18:534–552, 1985.

[114] S. Miyazawa and R. L. Jernigan. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term for simulation and threading. *J. Mol. Biol.*, 256:623–644, 1996.

[115] J. H. A. Nagel and C. W. A. Pleij. Self-induced structural switches in RNA. *Biochimie*, 84:913–923, 2002.

[116] A. Nahvi, N. Sudarsan, M. S. Ebert, X. Zou, K. L. Brown, and R. R. Breaker. Genetic control by a metabolite binding mRNA. *Chem. Biol.*, 9:1043–1049, 2002.

[117] R. Nussinov, G. Piecznik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matching. *SIAM J. Appl. Math.*, 35:68–82, 1978.

[118] H. Nymeyer, A. E. Garcia, and J. N. Onuchic. Folding funnels and frustration in offf-lattice minimalist protein landscapes. *Proc. Natl. Sci. USA*, 95:5921–5928, 1998.

[119] R. L. Ornstein, R. Rein, D. L. Breen, and R. D. MacElroy. An optimized potential function for the calculation of nucleic acid interaction energies. I. base stacking. *Biopolymers*, 17:2341–2360, 1978.

[120] A. T. Perrotta and M. D. Been. A toggle duplex in hepatitis delta virus self-cleaving RNA that stabilizes an inactive and a salt-dependent pro-active ribozyme conformation. *J. Mol. Biol.*, 279:361–373, 1998.

[121] J. Ponder and D. A. Case. Force fields for protein simulation. *Adv. Prot. Chem.*, 66:27–85, 2003.

[122] R. A. Poot, N. V. Tsareva, I. V. Boni, and J. van Duin. RNA folding kinetics regulates translation of phage MS2 maturation gene. *Proc. Natl. Acad. Sci. USA*, 94(19):10110–10115, 1997.

[123] S. B. Prusiner. The prion diseases. *Sci. Am.*, 272:48–61, 1995.

[124] H. Putzer, N. Gendron, and M. Grunberg-Manago. Co-ordinate expression of the two threonyl-tRNA synthetase genes in Bacillus subtilis: Control by transcriptional antitermination involving a conserved regulatory sequence. *EMBO J.*, 11:3117–3127, 1992.

[125] D. C. Rapaport. On three-dimensional self-avoiding walks. *J. Phys. A: Math. Gen.*, 18:113–126, 1985.

[126] J. Reeder and R. Giegerich. Improved efficiency of RNA secondary structure prediction in cluding pseudoknots. *unpublished*, ECCB 2002 poster; http://bibiserv.techfak.u ni-bielefeld.DE/pknotsrg/, 2002.

[127] C. M. Reidys and P. F. Stadler. Combinatorial landscapes. *SIAM Review*, 44:3–54, 2002.

[128] A. Renner and E. Bornberg-Bauer. Exploring the fitness landscapes of lattice proteins. In R. B. Altman, K. Dunker, L. Hunter, and T. E. Klein, editors, *Proc. 2nd. Pacif. Symp. Biocomp.*, Singapore, 1997. World Scientific.

[129] A. Renner, E. Bornberg-Bauer, I. L. Hofacker, P. K. Schuster, and P. F. Stadler. Self-avoiding walk models for non-random heteropolymers. Technical report, University Vienna, 1996.

[130] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction incuding pseudoknots. *J. Mol. Biol.*, 285:2053–2068, 1999.

[131] J. Ruan, G. D. Stormo, and W. Zhang. An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics*, 20:58–66, 2004.

[132] W. Saenger. *Principles of Nucleic-Acid Structure.* Springer-Verlag, New York, first edition edition, 1984.

[133] C. E. Schafmeister, S. L. LaPorte, L. J. W. Miercke, and R. M. Stroud. A designed four helix bundle protein with native-like structure. *Nature Struct. Biol.*, 4:1039–1046, 1997.

[134] J. A. Schellman. The factors affecting the stability of hydrogen-binded polypeptide structures in solution. *J. Chem. Phys.*, 62:1485, 1958.

[135] E. A. Schultes and D. P. Bartes. One sequence, two ribozymes: Implications for the emergence of new ribozyme folds. *Science*, 289:448–452, 2000.

[136] J. E. Shea and C. L. Brooks III. From folding theories to folding proteins: A review and assessment of simulation studies of protein folding and unfolding. *Annu. Rev. Phys. Chem.*, 52:499–535, 2001.

[137] J. E. Shea, Y. D. Nochomovitz, Z. Guo, and C. L. Brooks III. Exploring the space of protein folding Hamiltonian: The balance of forces in a minimalist β-barrel model. *J. Chem. Phys.*, 109:2895–2903, 1998.

[138] C. Simmerling, B. Strockbine, and A. E. Roitberg. All-atom structure prediction and folding simulations of a stable protein. *JACS*, 124:11258–11259, 2002.

[139] J. Skolnick and A. Kolinski. Simulations of the folding of a globular protein. *Science*, 250:1121–1125, 1990.

[140] N. D. Socci and J. N. Onuchic. Folding kinetics of proteinlike heteropolymers. *J. Chem. Phys.*, 101:1519–1528, 1994.

[141] A. D. Sokal. Monte Carlo Methods for the Self-Avoiding Walk. In Kurt Binder, editor, *Monte Carlo and Molecular Dynamics Simulations in Polymer Science*, chapter 2. Oxford University Press, 1996.

[142] G. Song, S. Thomas, K. A. Dill, M. Scholtz, and N. M. Amato. A path planning-based study of protein folding with a case study of hairpin formation in protein G and L. In *Pac Symp Biocomput.*, pages 240–251, 2003.

[143] P. F. Stadler. Towards a theory of landscapes. In R. Lopéz-Peña, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertuche, editors, *Complex Systems and Binary Networks (Proceeding of the Guanajuato Lectures 1995)*, pages 77–163. Springer-Verlag, 1996.

[144] P. Stolorz. Recursive approaches to the statistical physics of lattice protein. In *Proc. 27th Hawaii International Conference on System Sciences*, 1994.

[145] M. Tacker, W. Fontana, P. F. Stadler, and P. Schuster. Statistics of RNA melting kinetics. *Eur. Biophys. J.*, 23:29–38, 1994.

[146] D. Thirumalai, S. A. Woodson N. Lee, and D. K. Klimov. Early events in RNA folding. *Annu. Rev. Phys. Chem.*, 52:751–762, 2001.

[147] C. Thurner, C. Witwer, I. L. Hofacker, and P. F. Stadler. Conserved RNA secondary structures in flaviviridae genomes. *J. Gen. Virol,*, 85:1113–1124, 2004.

[148] I. Tinoco Jr. and C. Bustamante. How RNA folds. *J. Mol. Biol.*, 293:271–281, 1999.

[149] A. Torcini, R. Livi, and A. Politi. A Dynamical Approach to Protein Folding. *J. Biol. Phys.*, 27:181–203, 2001.

[150] R. Unger and J. Moult. Finding lowest free energy conformation of a protein is an NP-hard problem: Proof and implications. *Bull. Math. Biol.*, 55:1183–1198, 1993.

[151] R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *J. Mol. Biol.*, 231:75–81, 1993.

[152] A. Šali, E. Shakhnovich, and M. Karplus. How does a protein fold? *Nature*, 369:248–251, 1994.

[153] A. Šali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding. *J. Mol. Biol.*, 235:1614–1636, 1994.

[154] W. F. van Gunsteren and H. J. C Berendsen. Groningen Molecular Simulation (GROMOS) Library Manual. Biomos, Nijenborgh 16, Groningen, NL, 1987.

[155] D. J. Wales, M. A. Miller, and T. R. Walsh. Archetypal energy landscapes. *Nature*, 394:758–760, 1998.

[156] A. E. Walter, D. H. Turner, J. Kim, M. H. Lyttle, P. Müller, D. H. Mathews, and M. Zuker. Co-axial stacking of helixes enhances binding of oligoribonucleotides and improves predicions of RNA folding. *Proc. Natl. Acad. Sci. USA*, 91:9218–9222, 1994.

[157] M. S. Waterman and T. F. Smith. RNA secondary structure: A complete mathematical analysis. *Math. Biosc.*, 42:257–266, 1978.

[158] H. Wegele, L. Müller, and J. Bruckner. Hsp70 and Hsp90 - a relay team for protein folding. *Rev. Physiol. Biochem. Pharmacol.*, 151:1–44, 2004.

[159] E. Westhof and L. Jaeger. RNA pseudoknots. *Current Opinion Struct. Biol.*, 2:327–333, 1992.

[160] W. Winkler, A. Nahvi, and R. R. Breaker. Thiamine derivatives bin messenger RNA directly to regulate bacterial gene expression. *Nature*, 419:952–956, 2002.

[161] M. T. Wolfinger. The Energy Landscape of RNA Folding. Master's thesis, University Vienna, 2001.

[162] M. T. Wolfinger, W. A. Svrcek-Seiler, Ch. Flamm, I. L. Hofacker, and P. F. Stadler. Efficient computation of RNA folding dynamics. *J. Phys. A: Math. Gen.*, 37:4731–4741, 2004.

[163] M. Wu and I. Tinoco Jr. RNA folding causes secondary structure rearrangement. *Biochemistry*, 95:11555–11560, 1998.

[164] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structure. *Biopolymers*, 49:145–165, 1998.

[165] K. Yue and K. A. Dill. Forces of tertiary structural organization in globular proteins. *Proc. Natl. Acad. Sci. USA*, 92:146–150, 1995.

[166] H. Zamora, R. Luce, and C. K. Biebricher. Design of artificial short-chained RNA species that are replicated by Qβ replicase. *Biochemistry*, 34:1261–1266, 1995.

[167] G. Zifferer, M. Hofstetter, and O. F. Olaj. Monte Carlo simulation studies of the correlation between global size and helical structure in biopolymers. *J. Chem. Phys.*, 115(13):6236–6242, 2001.

[168] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[169] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull. Math. Biol.*, 46:591–621, 1984.

[170] M. Zuker and P. Stiegler. Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. *Nucl. Acids Res.*, 9:133–148, 1981.

Curriculum vitae

Mag. Michael Wolfinger
* 7. Juli 1976 in Linz, Oberösterreich

Bildungsweg

09/1982 – 07/1986	Volksschule Goethestraße, Linz
09/1986 – 06/1994	AHS Kollegium Aloisianum, Linz
06/1994	Matura
10/1994 – 06/1995	Präsenzdienst beim FlHB3, Hörsching, Oberösterreich
10/1995 – 03/2001	Diplomstudium: Chemie, Universität Wien Diplomarbeit am Institut für Theoretische Chemie und Molekulare Strukturbiologie bei Prof. Peter Stadler *The Energy Landscape of RNA Folding*
03/2001 – 10/2004	Doktoratsstudium: Chemie, Universität Wien Dissertation am Institut für Theoretische Chemie und Molekulare Strukturbiologie bei Prof. Peter Stadler *Energy Landscapes of Biopolymers*