Exploration of biopolymer energy landscapes via random sampling

Diplomarbeit
zur Erlangung des akademischen Grades
Diplom-Bioinformatiker

Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik

eingereicht von Andreas Stefan Richter
geboren am 11. April 1983 in Görlitz

Gutachter:	Prof. Dr. Rolf Backofen
	Prof. Dr. Stefan Schuster
Betreuer:	Prof. Dr. Rolf Backofen
	Dr. Sebastian Will

Jena, 19. Juli 2007

Besonderer Dank gilt

Rolf Backofen, Stefan Schuster, Sebastian Will.
Stefan Beyer, Conny Delor, Cathleen Dick, Mario Fasold, Michael Hecker, Alexander Heidrich, Christian Komusiewicz, Martin Mann, Anne-Marie O'Neill, Michael Wolfinger.
Michael Richter, Bärbel Richter, Helmut Richter.

Zusammenfassung

RNA und Proteine sind zwei bedeutende Biopolymere. Es wird allgemein angenommen, dass die Strukturen von RNA-Molekülen und Proteinen eindeutig durch deren Sequenzen bestimmt werden. Die Struktur eines Biomoleküls ist wiederum zur Ausübung seiner biologischen Funktion notwendig. Durch diskrete Strukturmodelle, welche eine abstrakte Beschreibung der molekularen Struktur geben, werden theoretische Studien am Computer ermöglicht. In dieser Arbeit wurden RNA-Moleküle als RNA-Sekundärstrukturen und Proteine als Strukturen auf einem Gitter repräsentiert. Der Strukturbildungsprozess von Biopolymeren wird entscheidend durch die Eigenschaften und die Topologie der Energielandschaft, welche der Faltung zu Grunde liegt, bestimmt. Typische Eigenschaften der Energielandschaften, wie beispielsweise die Anzahl der lokalen Optima, die Verteilung der Basins, aber auch die Übergangszustände zwischen den Optima, können hervorragend durch sogenannte Barrier-Trees veranschaulicht werden. Barrier-Trees bieten eine reduzierte Darstellung der Energielandschaft und ermöglichen dadurch die Untersuchung der Faltungsdynamik von Biopolymeren.

Im Rahmen dieser Diplomarbeit wird ein generischer, problemunabhängiger Ansatz zur Berechnung von Barrier-Trees vorgestellt. Im Gegensatz zu vorherigen Ansätzen basiert er nicht auf vollständiger oder teilweiser Aufzählung von Strukturen, da diese Methoden aufgrund des begrenzt verfügbaren Speichers auf kleine Moleküle beschränkt sind. Um eine gute Annäherung für den Barrier-Tree der Energielandschaft zu erhalten, wurden zufällige Wege zwischen lokalen Minima gesucht. Dieses Durchsuchen des Konformationsraumes wird als Sampling bezeichnet. Die dadurch über die Energielandschaft gewonnenen Informationen wurden genutzt, um den Barrier-Tree parallel zum Sampling zu konstruieren. Ansätze, welche auf Aufzählung basieren, erlauben es, den exakten Barrier-Tree für Moleküle beschränkter Größe zu berechnen. Um die hier vorgestellte Methode zu überprüfen, wurden für solche kleinen Beispiele die mittels Sampling berechneten mit den exakten Barrier-Trees verglichen.

Zwei Beispiele von RNA-Molekülen ergaben eine vollständige Übereinstimmung zwischen den berechneten Barrier-Trees. Dies deutet darauf hin, dass mit dem Sampling-Ansatz sowohl alle lokalen Minima als auch der exakten Barrier-Tree einer Energielandschaft berechnen werden können. Zwei Beispiele von Gitter-Proteinen zeigten, dass durch die in dieser Arbeit vorgestellte Methode der untersuchte Konformationsraum nicht auf bestimmte Regionen beschränkt wird. Es wurden mehr lokale Minima als in vorherigen Studien gefunden und die erhaltenen Barrier-Trees deckten einen größeren Bereich der Energielandschaft ab. Die erlangten Ergebnisse weisen darauf hin, dass eine Strategie, welche teilweise Aufzählung und Sampling kombiniert, die besten Ergebnisse versprechen würde.

Abstract

The structures of RNA molecules and proteins, which are both important biopolymers, are commonly assumed to be uniquely determined by their sequences. The structures of these biomolecules are in turn necessary to carry out the molecules' biological functions. Discretized structure models provide a coarse-grained description of the molecular structure, which is necessary to perform computational studies. In this research, RNA molecules were modeled as secondary structures for RNA, and proteins were modeled as self-avoiding walks on a lattice. The structure formation process of biopolymers is crucially determined by the properties and the topology of the underlying energy landscape, in which the folding proceeds. Typical characteristics of the energy landscape, like the number of local optima, the basin distribution as well as the transition states between the optima, can be visualized by barrier trees. Barrier trees provide a reduced representation of energy landscapes, which can be used to study the dynamical behavior of biopolymer folding.

The research described in this thesis aimed to present a generic, problem-independent approach for the generation of barrier trees. In contrast to previous studies, the approach used did not rely on exhaustive or selective enumeration, which is limited to smaller molecules due to the amount of available memory. In order to find a good approximation for the barrier tree of the energy landscape, walks between local minima were sampled by random and adaptive walks. The information determined about the energy landscape was used to build up the barrier tree during the sampling. Approaches which are based on enumeration allow to compute the exact barrier tree of an energy landscape for limited molecule sizes. To validate the presented method, the barrier trees resulting from the sampling were compared with the exact ones for such small instances.

Total agreement between the resulting barrier trees was obtained for two examples of RNA molecules, which indicates that the sampling approach can be used to compute both all local minima and the exact barrier tree of an energy landscape. Two examples of lattice proteins showed that the presented method does not restrict the investigated conformation space of the energy landscape to certain regions. More local minima than in previous studies were found, and the resulting barrier trees covered a larger region of the energy landscape. The results suggest that a strategy which combines selective enumeration and sampling for the exploration of energy landscapes promises the best results.

Contents

Chapter 1

Introduction

1.1 Motivation

Molecules, which are built of a large number of small subunits, are called polymers. The term monomer denotes the subunit of such a macromolecule. If the polymers are produced by living organisms, they are called biopolymers. Important biological macromolecules are the nucleic acids DNA (deoxyribonucleic acid) and RNA (ribonucleic acid), as well as proteins. Their monomers are nucleotides and amino acids, respectively. This thesis focuses on RNA and proteins, whose three-dimensional structure is vital for their biological function. We discuss approaches for the exploration of energy surfaces which govern the structure formation process of RNA and proteins.

In general, the specific sequence of amino acids and nucleotides uniquely determines the structure of a protein and a RNA molecule, respectively. The function of proteins and RNA in turn is determined by their structure.

For a long time, proteins have been known to show a great variety of three-dimensional structures. These structures are essential to carry out the protein's very manifold structural and functional roles in the cell.

In contrast, it was believed for decades that RNA molecules are little more than a simple carrier for information from DNA to proteins, since an important role of RNA molecules is to guide the polymerization of proteins. Nevertheless, numerous other functions of these molecules were revealed during the last few years. The Science Magazine decided that the discovery of special small RNAs, which operate many of the cell's controls, deserves the trophy "Breakthrough of the year 2002" [Cou02]. In 2006, even the Nobel prize for medicine was granted for the finding of gene silencing by RNA interference. However, crucial for the function of an RNA molecule is its secondary and tertiary structure. A well-known and well-studied example of the relation between structure and function is the tRNA. These are short sequences of about 76 nucleotides which form a cloverleaf secondary structure and typically occur in an L-shaped, tertiary structure. For further reading on the different types of RNA and their related structures, see for example [Hig00].

Due to the fact that the structure of a biopolymer is responsible for its function, it is clear that there is a big interest in the three-dimensional shape of biomolecules.

Therefore, an important task is to gain more knowledge about the folding of a biomolecule.

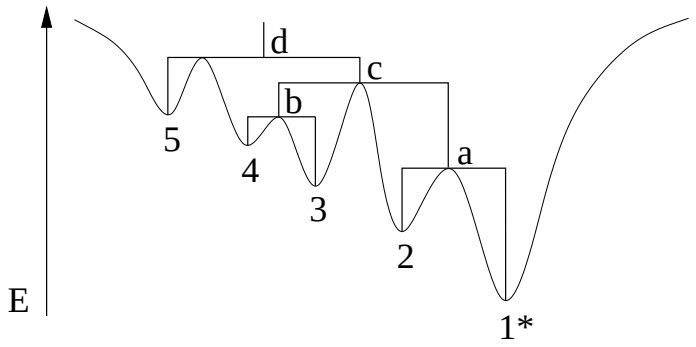


Figure 1.1: Schematic representation of an energy landscape and its associated barrier tree, taken from ref. [WWH+06]. The local minima are marked with numbers, and their connecting saddle points are marked with lowercase letters. The global minimum of the energy landscape is marked with an asterisk.

The term *folding* denotes the structural change from the open chain to the naturally occurring form. A typical requirement for the analysis of this process is to know the energetically optimal structures. They can be gained from a task called biopolymer structure prediction, which is the prediction of the biopolymer's structure from its sequence. Due to the very high number of degrees of freedom in an unfolded RNA or protein chain, the task of structure prediction is at present computationally feasible only in simplified models.

An impression of possible pathways from the unfolded to the stable ground state can be received from the study of the energy surface, on which the folding proceeds. The formation of a biomolecule's structure is crucially determined by the properties and the shape of its underlying folding landscape. These energy landscapes exhibit the same geometrical features as natural occurring landscapes, like mountains, valleys, plains, ridges and so on, but they are multidimensional. Typical characteristics of a landscape, like the number of local optima, the basin distribution as well as the transition states between the optima, can be conveniently visualized in the manner of a barrier tree. These barrier trees provide a reduced representation of energy landscapes, and they are a very useful description for the study of biopolymer folding pathways [FFHS00]. They also give an appropriate impression of the overall topology of the energy landscape. Having the underlying energy landscape of a biopolymer at hand, the folding dynamics of the molecule can be investigated, and properties of the folding landscape like kinetic traps can be unveiled easily. See Figure 1.1 for a schematic energy landscape representation and the associated barrier tree.

1.2 Related Work

Several studies that address the folding of biomolecules in combination with their underlying energy landscapes were carried out in the past.

Flamm et al. presented a stochastic algorithm named `kinfold` to simulate the folding kinetics of RNA sequences into secondary structures [FFHS00]. The correlation between the obtained kinetics and the structure of the energy landscape as well as folding mechanisms were discussed. In the past, it was often conjectured that a large free energy gap between the biopolymer's ground state and its first suboptimal structure indicated good folding properties. However, it turned out that this conjuncture is incorrect, at least for RNA secondary structures. Instead, Flamm et al. showed that the numbers and the heights

of saddle points along the folding path from the open chain to the folded structure are important factors which determine the folding behavior. Barrier trees, which organize the local minima and the saddle points in a hierarchical structure, therefore are an excellent tool for the study of folding pathways. However, the stochastic simulation used in their study considers all legal biopolymer structures, which makes it very time-consuming and computationally intensive.

Since there is a clear relationship between the dynamics and the energy landscapes of biopolymer folding, a formal definition of barrier trees and the associated basin structure in arbitrary landscapes had to be given. In [FHSW02], Flamm et al. developed a rigorous concept of barrier trees for degenerate landscapes. Furthermore, the program `barriers`, which efficiently computes the barrier tree of an energy landscape from an energy sorted list of configurations, was presented. `barriers` is based on exhaustive enumeration of all configurations. Therefore, its use is limited to landscapes of modest size. The program was applied to two well-known examples of landscapes: spin glass and RNA.

Thereinafter, it was shown that, based on the barrier tree approach, it is possible to predict the folding behavior of RNA molecules by numerical integration [WSSF$^+$04]. In their study, it was found out that there is a good agreement between the results of stochastic folding simulations and the dynamics predicted with the help of barrier trees.

Just recently, a generic algorithm to generate and explore the lower part of energy landscapes was presented and applied to discrete protein models [WWH$^+$06]. Given a starting set of low energy structures, the `latticeFlooder` approach allows to investigate parts of the energy spectrum selectively, restricted by a given energy threshold. It is possible to generate just the lower portion of the energy landscape, which contains the structures of main interest, like the optimal and suboptimal structures. A straightforward application of this method is the calculation of barrier trees. However, the method enumerates only structures below a given energy threshold and is limited by the available memory. The resulting barrier trees represent just a partial landscape.

Wang and Landau introduced a self-learning Monte-Carlo algorithm that performs a random walk on the energy surface [WL01a, WL01b]. This method allows to iteratively calculate the density of states, which is basically the number of states at a certain energy. Since the Wang-Landau algorithm is not trapped by local minima, it is suitable to efficiently sample rough energy landscapes. A modified version of this algorithm was proposed and implemented by Rathore et al. to study the folding of proteins [RdP02, RIdP03, RIdP06].

A completely different approach for studying protein folding was presented by Song et al. [STD$^+$03]. Their method evolved from robotics motion planning techniques, which are called probabilistic roadmap methods. Their approach aimed to study issues related to the folding process like the formation of secondary and tertiary structure, and the dependence of the folding pathway on the initial denatured structure. Since the roadmaps contain large sets of unrelated folding pathways, they provide global information about the protein's energy landscape.

1.3 Contribution

To gain knowledge about the energy landscape underlying a biopolymer folding process, exhaustive enumeration of all possible biopolymer structures can be used as a basis for

the calculation of the exact barrier tree. This was done in previous studies, see for example [FHSW02]. However, the search space grows exponentially with the length of the molecule, even in simplified models [MS96, Wat95]. Due to the immense number of possible structures, such exhaustive enumeration is obviously both time-consuming and hardly computationally feasible, in particular because of the limited amount of available memory. The selective enumeration approach presented by Wolfinger et al. [WWH$^+$06] results in barrier trees that represent just a cutout of the landscape.

This gives rise to the question whether it is necessary to enumerate all possible structures of a biopolymer in order to construct the barrier tree of its energy landscape, or if there are other possible approaches.

In this thesis, we enter this question and present a generic, problem-independent approach for the study of energy landscapes. Here, random sampling of the energy landscape is employed to find a good approximation of the landscape's barrier tree. The presented method by itself does not restrict the search space and enables, as a matter of principle, the approximation of the barrier tree with arbitrary accuracy. From the resulting barrier tree, one can derive topological details of the energy landscape: its shape, the number and the distribution of both global and local optima of the landscape, the basin structure and the heights of the barriers separating the optima. This information provides insight into the dynamical behavior of biopolymer folding.

The developed algorithm was applied to different examples of RNA and proteins in simplified models. The obtained results were afterwards evaluated by comparison with the results from previous studies [WSSF$^+$04, WWH$^+$06].

1.4 Overview

In Chapter 2, the theoretical background for the study of biopolymer energy landscapes is given. The chapter describes fundamental concepts of energy landscapes and simplified biopolymer structure models. Chapter 3 presents the sampling strategy used within this thesis and describes the implementation of the energy landscape models. In Chapter 4, examples of the sampling are given. Furthermore, the obtained computational results are presented and discussed. Finally, Chapter 5 recapitulates the results and gives an outlook of possible further research in this area.

Chapter 2

Fundamental Concepts and Definitions

In this chapter, the fundamental concepts needed for the purpose of this thesis are explained and several definitions are given. Thereby, the chapter provides a deeper introduction of biopolymers and their associated folding landscapes.

2.1 Energy Landscapes

In 1932, the idea of *fitness landscapes* was introduced in evolutionary biology to describe the dynamics of evolutionary optimization [Wri32]. This concept comprises both a set of genotypes arranged in an abstract space, which defines the accessibility of each genotype to another one, and a fitness value, which is assigned to each genotype by a *fitness function* f.

However, this concept is not only restricted to evolutionary processes. It is, for example, also used to model problems like combinatorial optimization where the fitness function is represented by the *cost function* [GJ79]. In biophysics, energy landscapes are used to describe the folding of biomolecules like proteins and nucleic acids. These folding landscapes, and especially their topology, are the main point of interest in this thesis[1].

A general definition of energy landscapes is given in [Sta02]. The following definition is a specialization for biopolymers:

An energy landscape can be described formally by the following three parts:

1. A set X of conformations, or more general configurations,
2. an operator $N : X \to \mathcal{P}(X)$, which defines the neighborhood of a conformation $x \in X$, and
3. an energy function $E : X \to \mathbb{R}$.

The conformation space $\mathcal{X}$ is formed by the conformation set X in combination with the neighborhood operator N. It can be distinguished between discrete landscapes, which

[1] To avoid confusion, it should be mentioned that in evolutionary context the fitness is maximized, and that in biophysics the energy is minimized.

have a finite conformation space, and continuous landscapes. In the following, only discrete landscapes will be discussed.

The organization of the conformation space $\mathcal{X}$ can be described by a *move set*. It defines how one conformation can be converted into a neighbored one [Sta02]. The move sets investigated here assign to each conformation $x \in X$ a set $N(x)$ of accessible neighbors. $N(x)$ denotes the *neighborhood* of x. Each move should have a reverse counterpart and the move set should be constructed such that $y \in N(x) \Leftrightarrow x \in N(y)$. The move set then results in a symmetric neighborhood relation $\mathfrak{N} : X \times X$, where $(x, y) \in \mathfrak{N} \Leftrightarrow y \in N(x)$. When applying a move to a string, which is a sequence of characters over a fixed alphabet like an RNA sequence, a character is typically replaced by another one at a single position.

An energy function E is called *non-degenerate*, if $E(\hat{x}) = E(\hat{y}) \Leftrightarrow \hat{x} = \hat{y}$. The definitions given in this section apply to non-degenerate energy landscapes. Distinctive features of degenerate landscapes will be specified separately.

The presented concept of energy landscapes allows the definition of local minima of the energy function, their associated basins of attraction and the saddle points and energy barriers separating them [Sta02]. The local minima are important in optimization problems since they might be decoys during the search for global minima. Global minima are minimal values of the energy function.

Formally, a conformation $\hat{x} \in X$ is called *local minimum*, or metastable state, if

$$\forall y \in N(\hat{x}) : E(\hat{x}) \leq E(y). \tag{2.1}$$

A conformation $\hat{x}$ is called a *global minimum*, if

$$\forall y \in X : E(\hat{x}) \leq E(y).$$

Note that each global minimum is a local minimum by definition.

The set of all local minima is denoted with $\mathcal{M}$. Moreover, let $\mathfrak{N}^+_{\mathcal{M}}$ be the transitive closure of $\mathfrak{N}$ on $\mathcal{M}$, i. e. the transitive closure of $\mathfrak{N}_{\mathcal{M}} = \{(x, y) \in \mathfrak{N} \mid x, y \in \mathcal{M}\}$. Then, $\mathcal{M}(x)$ is written for the set of local minima that are neighbored to $x \in \mathcal{M}$ directly or via other minima, i. e. $\mathcal{M}(x) = \left\{y \in \mathcal{M} \mid (x, y) \in \mathfrak{N}^+_{\mathcal{M}}\right\}$. Because the energy function E is, of course, constant on $\mathcal{M}(x)$, the definition is relevant for the degenerate case only. $\mathcal{M}(x)$ is called a *shoulder*, if

$$\exists z \in X \setminus \mathcal{M} \wedge \exists y \in \mathcal{M}(x) \text{ with } (y, z) \in \mathfrak{N} \text{ such that } E(y) = E(z). \tag{2.2}$$

This definition is illustrated in Figure 2.1. Confer [FHSW02] for a definition on graphs.

An important characteristic of a landscape is the intuitive notion of its *ruggedness*. The number of local minima is a measure for it, and the difficulty of the optimization on a landscape is closely related to its ruggedness [MdWS91].

Each local minimum $\hat{x}$ has an associated basin $\mathcal{B}(\hat{x})$, which is a set of structures that are attracted by the minimum. Although, before this characteristic of a landscape can be defined, a few more definitions have to be given first.

A list of conformations

$$x = x_1, \ldots, x_k = y \text{ with } \forall 1 \leq i \leq k : x_i \in X \text{ and } \forall 1 \leq i < k : (x_i, x_{i+1}) \in \mathfrak{N}$$

is a *walk* between the conformations x and y.

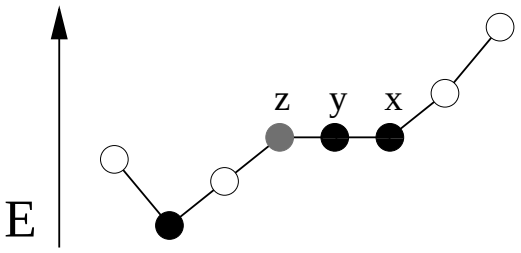


Figure 2.1: The shoulder, a special class of local minima. The black circles mark local minima. The minima x and y form a shoulder. The gray circle marked with z is a saddle point, but not a local minimum of the landscape.

The term *random walk* denotes an arbitrary, randomly chosen walk between two conformations. A walk is called an *adaptive walk*, if the conformations $x_1, \ldots, x_k$ hold the condition $\forall 1 \leq i < k : E(x_{i+1}) < E(x_i)$, and if $\nexists y \in N(x_k) : E(y) < E(x_k)$. On the other hand, a walk is called a *gradient walk*[2], if the conformations $x_1, \ldots, x_k$ hold the conditions $\forall 1 \leq i < k : E(x_{i+1}) < E(x_i) \wedge x_{i+1} = \underset{x \in N(x_i)}{\arg\min}\, E(x)$, and if $\nexists y \in N(x_k) : E(y) < E(x_k)$. That is, in each step of the gradient walk, the neighbor with the minimal energy has to be chosen. In degenerate landscapes, the minimum energy neighbor does not have to be uniquely defined. If several minimum energy neighbors exist, a deterministic rule can be used to choose the neighbor. For instance, one could always choose the neighbor which comes lexicographically first.

Each conformation $x \in X$ is mapped to a local minimum by performing a gradient walk from x. The *basin of attraction* of the local minimum $\hat{x}$, which is denoted by $\mathcal{B}(\hat{x})$, consists of all conformations which are mapped to $\hat{x}$ by a gradient walk. It should be noted that the mapping is unique in non-degenerate landscapes. In the degenerate case, the mapping becomes unique by the use of the aforementioned deterministic rule for choosing the minimum energy neighbor. The size of the basin and the fitness of the minimum are correlated: deeper minima usually have larger basins. A possible way to measure the size of a basin $\mathcal{B}(\hat{x})$ is to determine the average length of the gradient walks from $y \in \mathcal{B}(\hat{x})$ to $\hat{x}$ [Sta02].

The local minima and their basins of attraction are separated by saddle points and their corresponding energy barriers. Two conformations x and y in X are called mutually accessible at the level η, written

$$x \leftrightarrow\!\!\!\!\!\!\underline{\rho\ \ \eta}\!\!\!\!\!\!\rightarrow y,$$

if there is a walk $\mathbf{w}$ in X from x to y, such that $\forall z \in \mathbf{w} : E(z) \leq \eta$ [FHSW02]. The *saddle height* $E[\hat{x}, \hat{y}]$ between two local minima $\hat{x}$ and $\hat{y}$ is the minimum height which makes them accessible from each other, that is

$$E[\hat{x}, \hat{y}] = \min\left\{\max\left[E(s) | s \in \mathbf{w}\right] \,\middle|\, \mathbf{w} : \text{walk from } \hat{x} \text{ to } \hat{y}\right\} = \min\left\{\eta \,\middle|\, \hat{x} \leftrightarrow\!\!\!\!\!\!\underline{\rho\ \ \eta}\!\!\!\!\!\!\rightarrow \hat{y}\right\}. \quad (2.3)$$

A point $s \in X$ that satisfies the condition (2.3) is called a *saddle point* between $\hat{x}$ and $\hat{y}$. In non-degenerate landscapes, each saddle point s connecting $\hat{x}$ and $\hat{y}$ with $E(s) = E[\hat{x}, \hat{y}]$ is unique.

The saddle heights $E[\hat{x}, \hat{y}]$ have the property to be an ultrametric distance measure on the set of the local minima, which is discussed for example in [RTV86, MH98]. That is to say,

[2] The gradient walk, in this case, is also called steepest descent walk, since the steepest descent algorithm is applied.

the saddle heights satisfy the condition

$$E[\hat{x}, \hat{y}] \leq \max\left(E[\hat{x}, \hat{z}], E[\hat{y}, \hat{z}]\right) \quad \forall \hat{x}, \hat{y}, \hat{z} \in \mathcal{M}. \tag{2.4}$$

The *barrier* of a local minimum is the height of the lowest saddle point which has to be overcome in order to reach a more favorable local minimum. The barrier $B(\hat{x})$, which encloses a local minimum $\hat{x} \in \mathcal{M}$, is in symbols

$$B(\hat{x}) = \min\left\{+\infty; E[\hat{x}, \hat{y}] - E(\hat{x}) \;\middle|\; \hat{y} \in \mathcal{M} : E(\hat{y}) < E(\hat{x})\right\}. \tag{2.5}$$

Another definition linked to saddle points is the *valley below the saddle* s, $\mathcal{V}(s)$ [WSSF+04]. The valley $\mathcal{V}(s)$ is a collection of conformations which are reachable from the saddle point s on a walk whose energy never exceeds the value $E(s)$. Therefore, all conformations in $\mathcal{V}(s)$ have an energy below or equal $E(s)$. Assume that two saddle points s and s' have the property $E(s') < E(s)$. The valley $\mathcal{V}(s')$ can then be either a subvalley of $\mathcal{V}(s)$, i. e. $\mathcal{V}(s') \subseteq \mathcal{V}(s)$, if $s' \in \mathcal{V}(s)$, or the valleys are disjoint, i. e. $\mathcal{V}(s') \cap \mathcal{V}(s) = \emptyset$, if $s' \notin \mathcal{V}(s)$. In consideration of the fact that saddle points separate local minima and that each valley by definition contains at least one saddle point, it follows that each valley contains (in non-degenerate landscapes at least two) local minima. Conversely, $\mathcal{V}(s) \subseteq \bigcup_{k:\, \hat{x}_k \in \mathcal{V}(s)} \mathcal{B}(\hat{x}_k)$, i. e. the valley $\mathcal{V}(s)$ is contained in the union of the basins $\mathcal{B}(\hat{x}_k)$ of the local minima $\hat{x}_k \in \mathcal{V}(s)$. It should be noted that $\mathcal{V}(s)$ contains only conformations with an energy below or equal $E(s)$, whereas the energy of the conformations in the basins $\mathcal{B}(\hat{x}_k)$ might exceed the value $E(s)$.

In degenerate landscapes, the question arises how to treat neighbored local minima with the same energy. When imagining a flat landscape, with the given definition every state is a local minimum. Therefore, in this study, the following strategy was embarked: if two local minima have the same energy, and if the difference between their saddle height and their energy is below or equal a given threshold ε, they shall belong to a common equivalence class, that is, they are assumed to be equivalent to each other. Formally, the equivalence class of a minimum $x \in \mathcal{M}$ is the set $[x] = \left\{y \in \mathcal{M} \;\middle|\; E(x) = E(y) \wedge E[x, y] - E(x) \leq \varepsilon\right\}$. The basin of the equivalence class $[x]$ is defined by $\mathcal{B}([x]) = \bigcup_{y \in [x]} \mathcal{B}(y)$. For the remainder of this thesis, when speaking about a minimum, the equivalence class of the minimum for degenerate landscapes is implied. Thus, the barrier height between two minima also applies to the members of their equivalence classes.

After all, the valleys, the local minima within them and the saddle points connecting the metastable states can be represented in a unique hierarchical structure. This hierarchical structure is called the *barrier tree* of the energy landscape. The barrier tree is a rooted graph $G(V, E)$. The vertex set V contains the local minima of the landscape and the saddle points connecting them. Each vertex has an associated energy value, which is the energy of the local minimum and the saddle height, respectively. The leaves of the tree are the local minima, and the internal nodes represent the saddle points. The energy barriers between two minima can be read off easily according to Definition (2.5)[3]. For a rigorous mathematical definition of barrier trees of degenerated landscapes, see [FHSW02].

An example of a barrier tree with a schematic representation of the underlying energy landscape is given in Figure 1.1. In this example, the local minima marked with the

[3]It is also possible to represent the barrier tree as a rooted and weighted graph. Then, instead of providing the energy value of each vertex, each edge is weighted with the energy barrier of the vertices which it connects.

numbers 2 and 3 are accessible to each other by the saddle point c. The saddle height $E[2,3]$ corresponds to $E(c)$. The energy barrier of the minimum 3 is $E(c) - E(3)$.

2.2 Discrete Models of Biopolymer Structures

The two types of biopolymers examined here, namely RNA molecules and proteins, both have distinct three-dimensional structures. These structures give the polymers specialized biochemical capabilities like the catalytic mechanisms of enzymes and ribozymes. A great number of biopolymer structures have already been revealed at full atomic resolution by structural biologists with the help of techniques such as nuclear magnetic resonance (NMR) spectroscopy and X-ray crystallography. Still, the amount of data collected in structural data banks is continuously increasing. The Protein Data Bank (PDB), a worldwide repository of three-dimensional structural data of large biological molecules, in particular proteins and nucleic acids, is a good illustration of this fact [BWF+00]. The PDB was established in 1971 with just 7 available structures. By the end of 2006, 40839 structures were deposed, but the number of structures in the PDB is still undergoing an approximate exponential growth.

However, some applications of structural data in biology, like studies of molecular evolution, do not necessarily require a description of the molecular structure on a *fine-grained* level, which is a list of the three-dimensional coordinates of each atom. To reduce the level of detail, low-resolution or *coarse-grained structure models* were introduced. In the simple *discretized structure models* presented here, each monomer is modeled by a single point or letter. A review of different models can be found in [SS04]. These models do not only have the advantage that they concentrate on the problem-specific basic features of the structure, but they can also be enumerated and are easier to handle when performing computational studies.

2.2.1 RNA

Biological Background

RNA is a single-stranded molecule, which is made from monomers that are called nucleotides. Each nucleotide consists of a sugar (ribose) with an attached phosphate group and a nitrogen-containing sidegroup: a base. The base may be either adenine (A), cytosine (C), guanine (G) or uracil (U). The sugars are linked to each other by phosphodiester bonds. The resulting polymer chain is formed by the sugar-phosphate backbone and the bases which protrude from it.

Since the RNA is single-stranded, its backbone is flexible which allows the polymer chain to bend back and to form hydrogen bonds with another part of the same strand. The base A can pair with its complementary base U, and C can pair with G. Apart from these standard, or Watson-Crick base pairs, other non-standard types like G pairing with U can be found occasionally. RNA chains can fold up in a variety of different shapes. The complementary base-pairings cause that the folding of an RNA molecule is determined by its nucleotide sequence. The resulting structures of the folded RNA molecules can give rise to their biological functions. An example for the relationship between structure and

function is the ribozyme. The catalytic activity of this RNA molecule is enabled by its specific structure (see for instance [FDZD98, LS94]).

RNA Secondary Structures

In this study, an RNA secondary structure model at a coarse-grained level is used instead of the spatial coordinates. Each nucleotide is represented by a single letter. Only the covalent bonds between consecutive nucleotides, hence the RNA sequence, and the non-covalent hydrogen bonds, the base-pairs, are considered. In the following, a formal definition of an RNA structure according to [WS78] will be given:

Let $s \in \{A, C, G, U\}^*$ be a sequence. Then, an *RNA structure* over s is formally defined as a set P of pairs,

$$P = \left\{ (i,j) \middle| \begin{array}{l} i < j \wedge s_i, s_j \text{ form a Watson-Crick} \\ \text{or a non-standard base pair (G-U)} \end{array} \right\}.$$

Any two base pairs (i, j) and $(k, l) \in P$ have to fulfill the two following conditions:

- $i = k \Leftrightarrow j = l$ since each base can pair with one other base at most, and
- $j < k$, $l < i$, $i < k < l < j$ or $k < i < j < l$ must be satisfied.

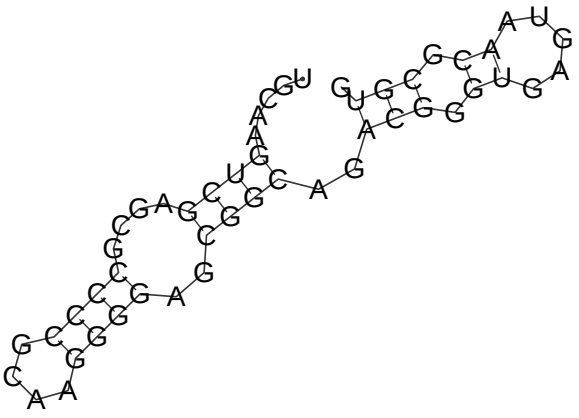
A structure satisfying the second condition is called non-crossing and does not contain pseudo-knots. After the formation of secondary structure elements, pseudo-knots as part of the tertiary structure can fold. They give rise to crossing base pairs, which is the reason why an RNA structure containing pseudo-knots is also called crossing. Pseudo-knots can be found in functionally important locations, which makes them important for many natural RNAs [tDPD92]. Since it is proved that the prediction of general RNA structures containing pseudo-knots is **NP**-complete [LP00], they are neglected in the remainder of this thesis.

As shown in Figure 2.2, an RNA secondary structure can be visualized as a planar secondary structure graph or, in a straightforward way, as a string in the bracket notation. For a structure on a sequence of length n, this string has the same length n and consists of dots and matching brackets. A base pair between the positions i and j is represented by a left parenthesis at position i and by a right parenthesis at position j. A dot stands for an unpaired base.

At this point, it can be returned to energy landscapes to define their abstract parts for RNAs. The conformation set X of a given RNA sequence s is the set of all secondary structures P, or conformations, which are compatible with s. The neighborhood of a conformation $x \in X$ is defined by a set of moves on x. The most elementary move set on an RNA secondary structure is the single move set. A single move assigns a structure $P_x \in X$ a neighbor $P_y \in X$ by removal or insertion of a single base pair (i, j) in compliance with the restriction that no pseudo-knots are allowed. The single move set always makes it possible to find a path[4] between two arbitrary conformations $P_x, P_y \in X$. The path can be constructed by the removal of all base pairs from P_x and the insertion of all base pairs from P_y into the unfolded intermediate structure afterwards. Due to the property that each two elements of X can be connected by a path, single moves provide an ergodic[5]

[4] A *path* (or *folding path)* is a walk where all conformations are distinct.

[5] *Ergodic* means that each arbitrary state of the conformation space must be reachable from each other state by the application of a finite number of operations from the move set.



.....((((....((((....))))..))))..(((.((.....)).))).

Figure 2.2: Visualization of an RNA secondary structure as RNA secondary structure graph and in bracket notation. Two matching parentheses symbolize a matching base pair, and unpaired bases are represented by dots.

move set on X. Therefore, it has been used in this study. For details about single moves and another class of moves on RNA secondary structures, the shift moves, see [FFHS00]. The single move set also induces a metric on the conformation space which is called the base pair distance.

The remaining part of the energy landscape, the energy function E, will be defined in the following section.

RNA Secondary Structure Prediction

The standard energy function of an RNA structure is assumed to be equal to the summed up energy contributions of all secondary structural elements, which the structure can be decomposed into:

$$E(P) = \sum_{(i,j)\in P} E^P_{i,j}, \tag{2.6}$$

where $E^P_{i,j}$ is the energy contribution of the secondary structural element defined by the base pair (i,j)[6]. These structural elements are hairpin and internal loops, bulges, stacked base pairs and multi-loops. Parameters for their energy can be found in [JTZ89, WTK+94, MSZT99], for example.

For the last two decades, several dynamic programming approaches to tackle the problem of RNA secondary structure prediction have been presented. Nussinov [NJ80] gave a very simple algorithm which assumes that the groundstate structure has the maximal number of base pairs. This "maximum matching" structure usually differs a lot from the real RNA structure. In 1981, Zuker and Stiegler [ZS81] formulated another algorithm to solve the RNA folding problem. It finds the structure with the *minimum free energy* (mfe), which is the thermodynamical most stable one, and uses the standard energy model of above. The mfe structure can be calculated recursively by dynamic programming. The algorithm is implemented, for instance, in the `Vienna RNA Package`[7] [HFS+94].

[6] Equation (2.6) is a simplification, since the energy contribution of dangling ends has to be added.

[7] The Vienna RNA Package is freely available at `http://www.tbi.univie.ac.at/RNA/`.

Another application of dynamic programming to the RNA folding problem was the observation made by McCaskill [McC90] that the partition function Z over all secondary structures P, which is

$$Z = \sum_{P \text{ is structure for } s} e^{-\frac{E(P)}{kT}}, \tag{2.7}$$

can also be calculated by dynamic programming. k is the Boltzmann constant, and T is the temperature. Having the partition function at hand, base pair probabilities and the probability of a given structure can be easily calculated.

An algorithm which generates all suboptimal conformations below a certain energy threshold was presented by Wuchty et al. [WFHS99]. The suboptimal folding is based on dynamic programming and multiple backtracking. The algorithm is part of the `Vienna RNA Package` as well. This approach allows to compute the *density of states* (DOS) in the low-energy region. The DOS is the distribution of the number of structures as a function of energy. The DOS is crucial to assess how well-defined the ground state is from a thermodynamical point of view.

2.2.2 Proteins

Biological Background

Like DNA and RNA, proteins are linear, unbranched chains composed of single monomers. In proteins, the monomers are amino acids and there are 20 different types of them[8].

All amino acids have the same general structure which is shown in Figure 2.3. An amino acid consists of a central carbon atom, the α-carbon (Cα) atom, and attached to it, the amino group (NH_2), the carboxyl group (COOH), a hydrogen atom (H) and the side-chain group (R). Each of the amino acids gets its unique properties from one of the 20 different side chains. They are different in regard to hydrophobicity, charge, reactivity, size, and so on.

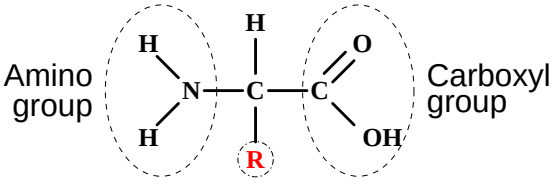


Figure 2.3: General structure of an amino acid, taken from ref. [BW06]. Attached to a central carbon atom are the amino group, the carboxyl group, the side-chain group (R) and an hydrogen atom.

The amino acids are linked together via covalent peptide bonds and form the polypeptide chain. A peptide bond connects the carboxyl group of one amino acid with the amino group of the next one. The order of the amino acids, which is called the *sequence* of the protein, is specific for each single protein.

[8] In human proteins, a 21st amino acid, called selenocysteine, is present. Unlike the 20 standard amino acids, it is not encoded directly by the genetic code.

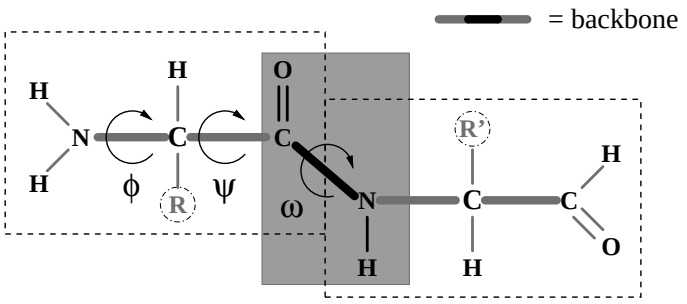


Figure 2.4: Two amino acids which are linked together by a peptide bond. The figure has been taken from ref. [BW06].

In Figure 2.4, it can be seen that the backbone[9] of the polypeptide contains three bonds per amino acid. The peptide bond is planar, that is, no free rotation around this bond is allowed. Two configurations are possible for the peptide bond: the *trans* form and the rare *cis* form (with a rotation angle ω of 180° and 0°, respectively). Rotation can occur around the Cα-C bond, which is called psi (ψ) angle, and around the N-Cα, which is called phi (ϕ) angle. The *conformation*, that is the arrangement of the atoms in the three-dimensional space, is determined by a pair of ψ and ϕ angles for each amino acid and the side chain angles. The two angles ψ and ϕ can both be in the range $[-180°, 180°]$. However, since steric collisions between the atoms must not occur, the possible angles are restricted to small regions. Nevertheless, the protein can still fold in a huge variety of ways and can therefore form an enormous amount of different conformations.

Each protein folds into a particular three-dimensional structure. Reactive sites on their surface allow the protein to bind with a high specificity to other molecules and to act as an enzyme to catalyze a reaction. Proteins also have other functions such as signal transduction, intracellular movement of other molecules, and the maintenance of cell structures. The specific function of each protein depends on its own specific amino acid sequence. Since the sequence is genetically specified, it could also be said that proteins put the genetic information of the cell into action. To delve further into biological details, consult for example [AJL$^+$02].

The folding of proteins takes place during and after their synthesis at the ribosome. The process happens spontaneously, but it is often assisted by special proteins which are called chaperons. It is assumed that the free energy of the conformation in functional proteins is minimized along the folding path until the *native structure* of the protein, the distinguished conformation of the natural protein, is reached. This whole process of searching for the native conformation is termed *protein folding*. However, it is possible that a misfolding occurs. For example, in the case of pathological proteins like prions, the folding stuck in a local minimum [HW97]. The treatment with special solvent can unfold, or denature, a protein by disruption of the noncovalent bonds which hold the protein into shape. After the removal of the denaturing solvent, the protein often renatures, that means it refolds into its native conformation. This strongly indicates that all the information needed for the protein folding process can be found in the amino acid sequence of the protein. For this reason, one tries to compute the native conformation just from the amino acid sequence, which is denoted *protein structure prediction*. Since the amount of available protein sequence data is growing enormously, but the experimental structure determination is very expensive

[9]The term *backbone* denotes the chain of Cα-atoms linked by the peptide bonds. The side-chain groups protrude from it.

and time-consuming, the number of known structures cannot keep up with the number of sequences. Therefore, protein structure prediction has become one of the most important problems in computational biology.

Simplified Models of Proteins

Lattice models provide a coarse-grained view on protein structure. They abstract the spatial coordinates of each amino acid to discrete positions on a given lattice and use a rather simple energy function. This allows to perform computational studies, which could not be realized with the full atomic resolution of proteins. The following definitions of simplified protein models are given in dependence on [Wil05].

The term *protein model* denotes a mathematical formalization to model the sequence, the structure and the energy of a protein. It consists of a set of sequences S over a fixed alphabet $\mathcal{A}$, a set of conformations (or structures) X and an energy function $E: \mathcal{A}^* \times X \to \mathbb{R}$ assigning an energy value to a sequence and a conformation.

A simple and well-known protein model is the *HP-Model* proposed by Lau and Dill in 1989 [LD89]. It reduces the 20-letter alphabet of the amino acids to a two-letter alphabet, consisting of H, which represents hydrophobic amino acids, and P, which represents polar or hydrophilic amino acids. Since it is commonly believed that the hydrophobic force is dominant in protein folding, the energy function only favors contacts between H-monomers. Only the backbone structure of the protein is modeled, that is one position for each amino acid. These positions are restricted to discrete positions on a geometrical structure that is known as lattice.

A *lattice* is a set L of lattice vectors (also called lattice points) such that

$$\begin{aligned} &\vec{0} \in L \text{ ($\vec{0}$ denotes the zero vector), and} \\ &\vec{u}, \vec{v} \in L \text{ implies } \vec{u} + \vec{v}, \vec{u} - \vec{v} \in L. \end{aligned}$$

For a lattice L, n vectors $\{\vec{v_1}, \ldots, \vec{v_n}\}$ exist in such a way that L consists of all the integral linear combinations of these vectors, that is

$$L = \left\{ \sum_{i=1}^{n} \lambda_i \vec{v_i} \;\middle|\; \lambda_1, \ldots, \lambda_n \in \mathbb{Z} \right\} \tag{2.8}$$

If n is minimal with the property (2.8), then $\vec{v_1}, \ldots, \vec{v_n}$ is a *basis* which generates the lattice L, and n is the *dimension* of L.

The Euclidean length of a lattice vector $\vec{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$ is $\sqrt{p_x^2 + p_y^2 + p_z^2}$ [10]. The non-zero lattice vectors with minimal Euclidean length are called *neighbor vectors* NV for L. Two lattice points $\vec{p}$ and $\vec{q}$ are called *neighbors* of each other, we say that they are in contact, if, and only if, $\vec{p} - \vec{q} \in NV$.

An example of a straightforward two-dimensional lattice ($\mathbb{Z}^2$) is the *square lattice* (`SQ`). The most simple three-dimensional one is the *cubic lattice* (`CUB`). Another lattice in $\mathbb{Z}^3$, the *face-centered cubic lattice* (`FCC`), is defined as the set of points

$$\left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{Z}^3 \;\middle|\; x + y + z \text{ is even} \right\}.$$

[10] This definition for three-dimensional lattices can be given analogously for two-dimensional lattices.

Name	Basis	Min. dist.	# Neighbors
SQ	$\{\binom{1}{0}, \binom{0}{1}\}$	1	4
CUB	$\left\{\begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}\right\}$	1	6
FCC	$\left\{\begin{pmatrix}-1\\-1\\0\end{pmatrix}, \begin{pmatrix}1\\-1\\0\end{pmatrix}, \begin{pmatrix}0\\1\\-1\end{pmatrix}\right\}$	$\sqrt{2}$	12

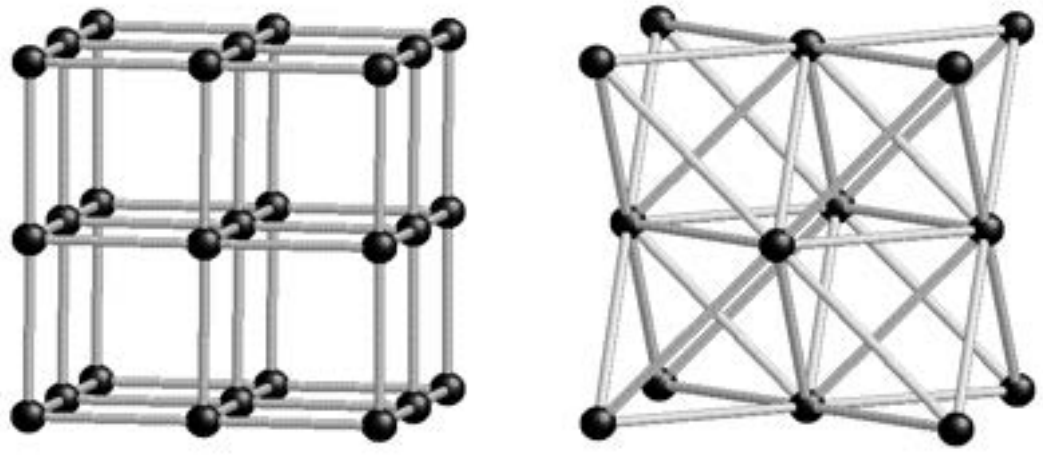


Figure 2.5: The cubic and the face-centered cubic lattice. Cutout of the cubic lattice with edges between neighbored points (left figure). The unit cell of the face-centered cubic lattice with edges between neighbors, in dependence on ref. [Wil05] (right figure).

Table 2.1 shows for these examples the generating basis vectors, the minimal Euclidean distance between two lattice points and the number of neighbors of each lattice point. A schematic representation of the cubic and the face-centered lattice can be found in Figure 2.5.

As it was shown in [PL95], the FCC lattice models real proteins structures more accurately than the cubic lattice. Another drawback of the latter lattice is known as the parity problem, which forbids contacts between points which have the same parity. The parity of a point is the sum of its coordinates which can either be even or odd. See [ABc+97] for details concerning the parity problem.

The abovementioned HP-model of Lau and Dill was originally introduced on the two-dimensional square lattice. It can be formally defined as a protein model with:

- a sequence $s \in \{H, P\}^n$
- a structure $x : [1, \ldots, n] \to \mathbb{Z}^2$ which fulfills the conditions
  1. $\forall 1 \leq i < n : x(i)$ and $x(i+1)$ are neighbors, and
  2. $\forall 1 \leq i < j \leq n : x(i) \neq x(j)$
- a contact energy function $E(s, x) = \sum_{1 \leq i < j \leq n} E_{s_i, s_j} \Delta\left(x(i), x(j)\right)$,

	H	P
H	-1	0
P	0	0

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

Table 2.2: Energy matrices for the pairwise contact potential E_{s_i,s_j} for different alphabets. **HP**: includes hydrophobic interaction (H = hydrophobic, P = polar) (left table). **HPNX**: includes hydrophobic and electrostatic interaction (H = hydrophobic, P = polar, N = negative, X = neutral), taken from ref. [BWBB99] (right table).

where $x(i)$ denotes the position of the i-th monomer of the structure x,

$$E_{s_i,s_j} = \begin{cases} -1 & \text{if } s_i = s_j = H, \\ 0 & \text{otherwise} \end{cases} \quad \text{and}$$

$$\Delta(p,q) = \begin{cases} 1 & \text{if } p \text{ and } q \text{ are neighbors,} \\ 0 & \text{otherwise.} \end{cases}$$

The first condition of the structure demands that successive amino acids of the chain are also neighbored in the lattice. The second condition ensures self-avoidance, that is none of the lattice positions is occupied by two monomers. The sequence of neighbored lattice points which describes the protein structure (first condition) is called a *walk* on the lattice. A walk which assures self-avoidance is a *self-avoiding walk* (SAW).

The presented HP-model can be extended in different ways. Although originally defined for the square lattice, the HP-model can be easily applied to other lattices. For example, two- and three-dimensional triangular lattices were used in [ABc+97]. It is also possible to extend the alphabet of the model. Energy matrices for the pairwise contact potential E_{s_i,s_j} for different alphabets can be found in Table 2.2.

Below, it will be shown how a walk on a given lattice can be encoded as a string. This compression of the structure has the advantage that the storage space is much smaller than in the case of saving the complete coordinates of each lattice point. Another benefit is that the comparison of structures can be reduced to simple string comparison. An introduction into this concept of absolute and relative moves was given by Bornberg-Bauer in [BB97]. Backofen et al. went further into the question in [BWC00]. An *absolute move* for a given lattice is a character of an alphabet $\mathcal{D}$ which is assigned to each possible neighbor vector of the lattice. For the square lattice, the directions forward, left, right and backward are described by $\mathcal{D}_{\mathtt{SQ}} = \left\{ d_1 = \binom{1}{0} := f, d_2 = \binom{0}{1} := l, d_3 = \binom{0}{-1} := r, d_4 = \binom{-1}{0} := b \right\}$. For a sequence of length n, a walk on a lattice is fully described by its initial point $x(0)$ and the ordered list of $n-1$ absolute moves. That is to say, the lattice point $x(i+1)$ is obtained by attaching the move $d \in \mathcal{D}$ to $x(i) : x(i+1) = x(i) + d$.

Relative directions provide an alternative in encoding the protein's conformation. This thesis follows the definition as given in [BWC00], which differs from Bornberg-Bauer's definition of relative moves [BB97]. A conformation of length n can be described by a walk of $n-1$ relative directions. Following along the walk in relative directions involves retaining a frame of reference, which is changed using rotation matrices.

In the following, the concept of relative moves is exemplified for the three-dimensional cubic lattice. In this lattice, it is necessary to apply a base transformation with every

relative move. A *relative move* is an element of the alphabet $\mathcal{R} = \{F, L, R, U, D, B\}$. In self-avoiding walks, the backwards direction B does not occur, which reduces the relative move alphabet to $\mathcal{R} \setminus B$. The vector v_r assigned to a relative move $r \in \mathcal{R}$ is defined as

$$v_F = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, v_L = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, v_R = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, v_U = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, v_D = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

A sequence $w \in \mathcal{R}^*$ is called a relative move sequence. For a given sequence w, $g_{baser}(w)$ is defined as

$$g_{baser}(w) = \begin{cases} I_3 & \text{if } w = \epsilon, \\ g_{baser}(w') \cdot B_r & \text{if } w = w'r. \end{cases}$$

I_3 denotes the 3×3 identity matrix, and B_r are rotation matrices which turn the vector v_r into $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. Thus, B_r is defined as follows:

$$B_F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, B_L = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, B_R = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, B_U = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, B_D = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}.$$

Let w be a given relative move sequence of length $n-1$. The lattice points of the corresponding conformation with the initial point $x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ are defined by $\forall 1 \leq i \leq n-1 :$ $x(i) = x(i-1) + g_{baser}(w_1 \dots w_{i-1}) \cdot v_{w_i}$.

The relative move sequences allow the introduction of pivot moves. A *pivot move* is a point mutation on the relative move string which corresponds to a rotation of the remaining conformation part behind the position it was applied to. In fact, pivot moves result not only in rotations, but they also allow reflections. Thus, mutations (and pivot moves) correspond to automorphisms which map the lattice to itself [BWC00]. The concept of pivot moves can be defined for arbitrary lattices. The ergodicity of pivot moves has been proved by Madras et al. [MS88].

Another class of moves applied to lattice proteins are *local moves*. A local move changes the positions of a bounded number of consecutive monomers at a time. The move set provided by this type of move is non-ergodic and was not taken into consideration in this study.

Finally, the parts of an energy landscape for lattice proteins are summarized. The conformation set consists of all self-avoiding walk structures which have the length of a given sequence s. The organization of the conformation space is described by the move set. In this case, the application of a single operation from the pivot move set, which is a point mutation on the structure in relative moves, generates the neighboring conformation of a given structure x. The definition of energy landscapes is completed with an energy function. In the lattice protein case, it is given by the sum of the pairwise contact potentials of the structure x.

Constraint-Based Protein Structure Prediction in Simplified Protein Models

Structure prediction in a given protein model can be regarded as the combinatorial optimization problem of predicting

$$\arg\min_{x \in X} E(s, x)$$

for a certain sequence s.

In 1998, it was shown that the problem of protein structure prediction is **NP**-complete, even in the HP-model for the two-dimensional [CGP+98] and cubic lattice [BL98]. Thus, there is probably no general, efficient algorithm that solves this problem (under the belief that $\mathbf{P} \neq \mathbf{NP}$).

In this section, a fast and exact approach for the prediction of optimal and suboptimal structures in simplified protein models on different lattices using the constraint programming technique is outlined. This method of Backofen and Will is termed *constraint-based protein structure prediction (CPSP)*. It is explained in detail in [BW06].

Overall, CPSP predicts optimal structures which have maximally compact hydrophobic cores. These compact hydrophobic cores are sets of points with as many contacts between H-monomers as possible. For the purpose of protein structure prediction, compact cores are enumerated in a process called core construction first. Subsequently, one tries to place the protein sequence on a compact core which is called threading.

Given a sequence s with n_H many H-monomers, CPSP starts with the computation of bounds on the number of possible HH-contacts for n_H monomers, if they are freely distributed to the lattice points. The main idea of this first step is to split the lattice into layers and to consider the contacts within and between the layers. In the following step of core construction, a constraint-based search for enumerating all cores of size n_H with maximally many contacts is performed. In the third and final step, the sequence s is mapped onto the cores. This threading is modeled as a constraint satisfaction problem. During the threading, one tries to find an optimal structure which fulfills two conditions: all H-monomers occupy core positions, and the structure is a self-avoiding walk, that is all positions differ from each other and the chain is connected. If it fails to map s onto an optimal core, the number of contacts will be relaxed and the threading is repeated until a structure is found. It should be noted that the first two steps described are not performed throughout each protein structure prediction, but that the optimal cores are precomputed independently of an actual sequence and stored in a data base. Only the threading step has to be carried out for each given sequence.

The CPSP approach outperforms other approaches for the folding of lattice proteins in HP-models concerning time efficiency, completeness and flexibility. The approach is complete since it predicts all optimal structures. It is flexible since it works for different lattices and contact energy functions, more precisely for the HP and HPNX-energy functions and for the `CUB` and the `FCC` lattice. CPSP was successfully applied for predicting optimal structures of HP-sequences up to a length of 200 in the face-centered cubic lattice, where only heuristic algorithms existed so far. Above all, CPSP is the only available method to completely enumerate the ground state and near-optimal states of lattice proteins in several three-dimensional models, and was therefore used in the context of this study.

2.3 Kinetics of Biopolymer Folding

In this paragraph, the kinetics of biopolymers will be discussed. The aim is to predict the folding behavior of biopolymers as a function of time.

A stochastic algorithm for the investigation of RNA folding kinetics was given by Flamm et al. [FFHS00]. In their contribution, the formation of RNA secondary structures was modeled at the level of single base pairing events like opening and closing of base pairs.

The following model was used:

Given the move set, the folding of biomolecules can be modeled as a continuous-time Markov process in conformation space as follows: Let X be a set of conformations that are compatible with the sequence s, in compliance with the prior definitions of RNA or lattice protein structures. The transition rate from the conformation $y \in X$ to the conformation $x \in X$ is given by r_{xy}. This rate is zero, if $(y, x) \notin \mathfrak{N}$. That is, y and x are not neighbors in the conformation space according to the defined move set. The probability of observing conformation x at time t as the secondary structure of s is denoted by $p_x(t)$. The probability distribution is given by the master equation

$$\frac{\mathrm{d}p_x(t)}{\mathrm{d}t} = \sum_{y \in X} p_y(t)\, r_{xy}, \text{ with } r_{xx} = -\sum_{y \neq x} r_{yx}.$$

The equation can be rewritten in matrix form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{p}(t) = \mathbf{R}\,\vec{p}(t). \tag{2.9}$$

This linear system of differential equations is solved by explicit computation of $\vec{p}(t) = e^{t\mathbf{R}}\,\vec{p}(0)$ with the initial distribution vector $\vec{p}(0)$. For the transition rates r_{yx} between neighboring structures, the model dictates the expression

$$r_{yx} = r_0 e^{-\frac{E^{\neq}_{yx} - E(x)}{kT}} \text{ for } x \neq y.$$

The transition state energies have to be symmetric: $E^{\neq}_{yx} = E^{\neq}_{xy}$. In the simplest case, they can be modeled by $E^{\neq}_{yx} = \max\{E(x), E(y)\}$. The time axis can be adjusted to experimental data with the parameter r_0.

However, the presented approach for the simulation of the whole kinetic folding process considers all possible biopolymer conformations. Since the conformation space grows exponentially with the sequence length [MS96, Wat95], this description of the energy landscape is computationally feasible only for very short sequences. Therefore, a coarse-grained representation of the energy landscape is needed, which brings us back to the concept of barrier trees. The barrier trees were introduced as a mapping from the full conformation space to a reduced conformation space, since they represent only the local minima and the saddle points of the folding landscape. Based on these, several discrete models can be formulated to predict the RNA folding behavior [WSSF$^+$04]. These models have been applied to lattice proteins as well [Wol04].

Consider the gradient basins $\mathcal{B}$ of the local energy minima to be a partition of the conformation space X. The classes of the partition are called *macrostates*. To each macrostate α, one can assign the partition function

$$Z_\alpha = \sum_{x \in \alpha} e^{-\frac{E(x)}{kT}}$$

and the corresponding free energy

$$G(\alpha) = -kT \ln Z_\alpha.$$

The simplest and most straightforward approximation for the kinetic folding process is the Arrhenius law for transitions on the barrier tree. Within this model, transitions only occur

between the local minima that are directly connected by a saddle point. The transition state energies are approximated by the saddle heights $E[\alpha, \beta]$. For the rates between macrostates α and β, one derives $r_{\beta\alpha} = e^{-\frac{E[\alpha,\beta]-G(\alpha)}{kT}}$. Instead of using the macrostate's free energy $G(\alpha)$ for the calculation of the transition rates, it is conceivable to use the energy of the local minimum belonging to the macrostate. However, this simplification lowers the quality of the approximation. A general drawback of the Arrhenius law is that it completely neglects the fact that there are multiple pathways connecting two minima.

A much better approximation for the kinetic folding process is the macrostates process. It calculates the transition rates between macrostates by summing up the microscopic rates between the conformations belonging to the macrostates. Since the `barriers` program determines to which macrostate a conformation belongs, the transition rates between the macrostates can be computed "on-the-fly" while executing the program.

A comparison between the folding dynamics resulting from the directly integrated master equation (2.9) and the coarse-grained dynamics shows reasonable agreement. For RNA, the Arrhenius law describes the process qualitatively correct, but differs clearly in quantitative details. The macrostates process exhibits better agreement to the stochastic simulations [WSSF$^+$04]. In contrast, coarse-grained lattice protein dynamics generally shows different behavior from the stochastic simulations of the kinetic folding process [Wol04].

In summary, barrier trees appear to be a good starting point for the calculation of folding kinetics. They provide a reduced representation of the conformation space, which is restricted to local minima and saddle heights. Based on these, Arrhenius-type kinetics can be formulated.

Chapter 3

Methods

The following chapter describes how the sampling approach of this study was carried out.

The pseudocode used in this thesis to specify the algorithms follows the conventions of Cormen et al. [CLRS01]. All presented algorithms were implemented in ISO/ANSI C++ using object-oriented programming.

3.1 Sampling of the Energy Landscape

In this study, the barrier tree of an energy landscape was constructed without exhaustive enumeration of all possible biopolymer structures. Instead, it was approximated by a sampling over the conformation space of the biopolymer. The performed sampling was not guided by a sophisticated heuristic, but was rather a randomized search. This allowed us to make as little assumptions of the underlying landscape as possible, and led to a generic approach. Therefore, the method could be applied to different systems which make use of the energy landscape concept.

As elucidated in Section 2.1, the barrier tree of an energy landscape is a rooted graph. The vertices of the graph correspond to the local minima of the landscape and their connecting saddle points. Each conformation which is represented by a vertex has an associated energy value. An energy function E assigns the energy to the conformations. Consequently, it is sufficient to know all local minima and their saddle points to construct the barrier tree of an energy landscape.

It should be noted that in the approach presented in this thesis, just the saddle height between two conformations, instead of the saddle point, was stored. This is due to two reasons: first, less storage space is needed. The second, but more important reason is that biomolecules in simplified models typically show degeneracy. A high degree of degeneracy is especially a common feature of lattice protein energy landscapes, since there are many conformations that have exactly the same energy. The saddle points connecting the optima do not have to be unique, but they must have the same energy. For this reason, we were only interested in the saddle heights which make two conformations mutually accessible. Moreover, two neighbored conformations which have the same energy are obviously mutually accessible at the level of their energy. Then, there is even no saddle point between them, but the saddle height is still defined according to Equation (2.3).

Whilst being aware of this, the following sampling strategy can be derived to construct the barrier tree iteratively:

1. While the sampling termination condition is not fulfilled, choose a minimum $\hat{x}$ out of all local minima that are already known.
2. Perform a random walk of length n, starting from the chosen minimum $x_1 = \hat{x}$. Save the end conformation x_n and the highest energy value E_{max} of all conformations that were visited during the random walk.
3. Perform an adaptive walk starting from the conformation x_n. The walk terminates in a local minimum $\hat{y}$.
4. If the minimum $\hat{y}$ is not yet known, add $\hat{y}$ to the barrier tree and connect it to $\hat{x}$ by the estimated saddle height E_{max}, since $\hat{x} \overset{E_{max}}{\leftrightarrow} \hat{y}$. If the minimum $\hat{y}$ is already known, and if E_{max} is lower than the current estimated saddle height E_{curr} between the minima $\hat{y}$ and $\hat{x}$ in the barrier tree, replace E_{curr} by E_{max}.
5. Iterate from step 1.

In the following subsections, the algorithms of the adaptive and the random walk are specified. In addition, a proof is given that the strategy presented above actually yields to local minima and the correct saddle heights between them.

Note that in the remainder of the thesis, the term saddle height is used for the estimated saddle height between two minima, that is the currently known energy that makes the minima mutual accessible to each other. Otherwise, the term correct saddle height is used.

3.1.1 The Adaptive Walk

Corollary 1. *Every adaptive walk terminates in a local minimum.*

Proof. According to the definition given in Section 2.1, an adaptive walk terminates in x_i, if $\forall x_{i+1} \in N(x_i) : E(x_{i+1}) \geq E(x_i)$. Hence, x_i is a local minimum according to Definition (2.1). □

The algorithm of the adaptive walk, starting from a given conformation x, is given in Listing 3.1.

Listing 3.1: Algorithm of an adaptive walk

```
ADAPTIVE_WALK(x)
1  while arbitrary neighbor y of x with E(y) < E(x) exists
2        do x ← y
3  return x
```

The function $E(x)$ implements the energy function E, which returns the energy value of the passed conformation x. Detailed information about the implementation of the conformation's neighborhood, the energy function and other properties of the biopolymer models is given in Section 3.5.

The application of a gradient walk to find unknown local minima is also possible. This walk characterizes the basins simultaneously, but it is not the objective here. The adaptive walk

has the advantage that, in contrast to the former, the algorithm is not bound to enumerate all neighbors of the current conformation. Thus, it is much faster and was therefore used in this approach.

3.1.2 The Random Walk

The random walk, starting from a given local minimum, aims to leave the basin of attraction of the minimum. With the subsequent adaptive walk, it is attempted to reach a conformation other than the start conformation.

The desired length n of the walk, that is the total number of visited conformations, has to be passed to the procedure `RANDOM_WALK` as given in Listing 3.2. Furthermore, it is assumed that the pointer to an object $WalkStatus$ is passed to the procedure. The object $WalkStatus$ is composed of the attributes x_curr for the currently processed conformation and e_max for the highest energy of all conformations that were visited during the random walk.

Listing 3.2: Algorithm of a random walk

```
RANDOM_WALK(WalkStatus, n)
1  e_max[WalkStatus] ← E(x_curr[WalkStatus])
2  i ← 1
3  while arbitrary neighbor x of x_curr[WalkStatus] exists and i ≤ n
4        do i ← i+1
5           if E(x) > e_max[WalkStatus]
6              then e_max[WalkStatus] ← E(x)
7           x_curr[WalkStatus] ← x
```

Proposition 1. *The sampling utilizing the presented random walk algorithm yields the correct saddle height between the start conformation of the walk, $\hat{x}$, and the local minimum $\hat{y}$, obtained by the subsequent adaptive walk, if the random walk length n has been chosen to be sufficiently large, and if a sufficient number of different walks from $\hat{x}$ to $\hat{y}$ have been found by the sampling.*

Let E_{max} be the highest energy that occurred during a random walk starting from $\hat{x} = x_1$. Further, let x_n be the conformation the random walk terminated in. Thus, $E(x_n) \leq E_{max}$. The subsequent adaptive walk starts from x_n and terminates in $\hat{y}$. Since only conformations with lower energy are chosen during each step of the adaptive walk (see Section 2.1), E_{max} is the highest energy on the performed walk $\mathbf{w}$ from $\hat{x}$ to $\hat{y}$. That is, $E_{max} = \max[E(x)|x \in \mathbf{w}]$, which means that x and y are mutually accessible at the energy level E_{max} for this walk.

According to Definition (2.3), the correct saddle height $E[\hat{x}, \hat{y}]$ is $\min\left\{\eta \,\middle|\, \hat{x} \overset{\eta}{\leftrightarrows} \hat{y}\right\}$, which is the minimum of all E_{max} found over all walks. As long as the value $E[\hat{x}, \hat{y}]$ was not found by the sampling, there must be at least another unknown walk v from $\hat{x}$ to $\hat{y}$, in which the highest energy value is equal to $E[\hat{x}, \hat{y}]$, such that $\hat{x} \overset{E[\hat{x},\hat{y}]}{\leftrightarrows} \hat{y}$. The repeated application of random and adaptive walks, starting from $\hat{x}$ and terminating in $\hat{y}$, yields new walks as long as the length n of the random walk has been chosen to be large enough. Due to the fact that the conformation space is finite, the walk v is likely to be found if a sufficient number of walks between $\hat{x}$ and $\hat{y}$ have been sampled.

3.1.3 The Sampling Approach

Corollary 2. *Once the set of all local minima $\mathcal{M}$ and the correct saddle heights between each possible pair of minima $x, y \in \mathcal{M}$ have been found with the presented strategy, the correct barrier tree of the energy landscape can be constructed from them.*

As stated above, this corollary follows from the definition of a barrier tree as given in Section 2.1.

Now, the whole sampling algorithm can be specified (see Listing 3.3). It is assumed that the passed parameter n is the number of single sampling steps and that m is the desired random walk length. A pointer to an existing barrier tree BT, which is not specified further at this point, has to be passed as well. This tree has to contain at least a single optimum to provide a start conformation for the sampling. The optima and the saddle heights which were found are added to BT during the sampling process.

In this study, local minima, which were part of a shoulder (see Definition (2.2)), were not regarded as local minima in the narrower sense. Thus, they were not included in the barrier tree, but they were still saved to avoid multiple processing. In the degenerate case, a local minimum that is part of a shoulder could be the endpoint of a gradient walk. This special case was not taken into consideration here. However, it could be reasonable to save these special local minima in the barrier tree.

Listing 3.3: Sampling algorithm

```
SAMPLING(BT, n, m)
 1  for i ← 1 to n
 2      do get local minimum x from BT
 3         x_curr[WalkStatus] ← x
 4         RANDOM_WALK(WalkStatus, m)
 5         y ← ADAPTIVE_WALK(x_curr[WalkStatus])
 6         if y is unknown local minimum
 7            then save y as known local minimum
 8                 if neither x nor y are part of a shoulder
 9                    then insert y as local minimum into BT
10                         add saddle height e_max[WalkStatus] between x and y in BT
11                    else if x is part of a shoulder
12                           then replace x by y in BT
13            else if y is in BT and e_max[WalkStatus] is lower than saddle height
                      between x and y in BT
14                    then update saddle height between x and y in BT
                           to value e_max[WalkStatus]
```

The energy landscape sampling approach is illustrated in Figure 3.1.

The resulting barrier tree is the correct one for the investigated energy landscape, if all local minima $\mathcal{M}$ and the correct saddle heights between all minima were found, as stated in Corollary 2. That is, this method is capable to yield the correct barrier tree unless the sampling was stopped before the correct tree has been found. In the latter case, a more or less good approximation of the barrier tree is obtained. Such an approximation can lack local optima, and the estimated saddle heights can be above the correct values.

The whole sampling process can be controlled by several parameters which have already been partially introduced. The length of the sampling is determined by the parameter n. Since the structure space grows exponentially with the chain length of a

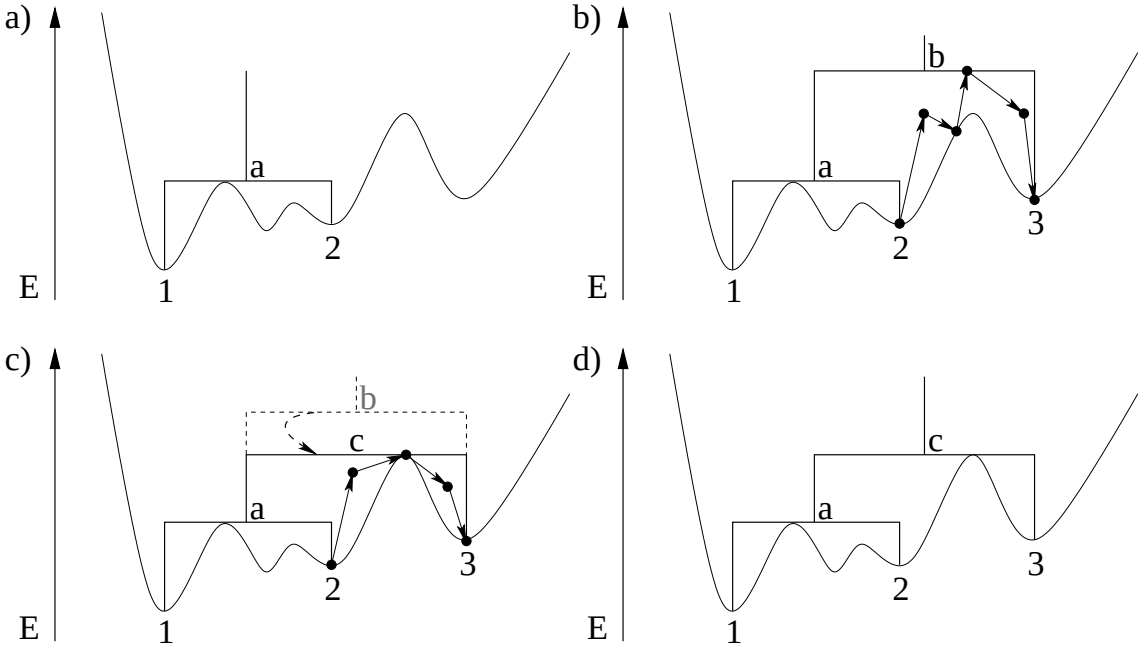


Figure 3.1: Sampling of the energy landscape. a) A barrier tree, which contains the two local minima 1 and 2, connected by the saddle point a, is given. b) The minimum 2 is randomly chosen as start conformation of a random walk. The subsequent adaptive walk, starting from the end conformation of the random walk, terminates in the local minimum 3. $E(b)$ is the highest energy value that occurred during the random walk. Since minimum 3 is not yet known, it is added to the barrier tree and connected to minimum 2 by the saddle point b with the height $E(b)$. c) Another sampled walk between minima 2 and 3 results in the saddle point c that connects 2 and 3. Since $E(c) < E(b)$, the estimated saddle height $E(b)$ between 2 and 3 is updated to $E(c)$, which is the correct saddle height. d) The resulting barrier tree of the energy landscape.

biomolecule[1], n should be chosen depending on the molecule's length.

Another alterable parameter is the length m of the random walk. If m is chosen too large, the sampling will be an arbitrary roaming in the structure space. Then, it will be unlikely to find the correct saddle height. On the other hand, by choosing the value of m too small, it cannot be assured that the basin of the start local minimum is left. To tackle this problem, the following strategy was mapped out to adjust the random walk length for a given problem instance: During the random walk, an adaptive walk was started from each processed neighbor. As soon as the adaptive walk terminated in a local minimum which has been unknown until then, the current length of the random walk was saved. This test was repeated many times. Then, the distribution of the random walk lengths was used to estimate an appropriate value for the parameter m.

Since each sampling iteration starts from a given conformation, the sampling process is, of course, also influenced by the method through which the start conformation is chosen from the barrier tree. The easiest idea is to pick it out randomly of all local minima already known under the premise that all optima are uniformly distributed.

However, the biologically functional structures are the energetically optimal and near-

[1] The number of RNA secondary structures, S_N, grows exponentially with the chain length N: $S_N \sim N^{-\frac{3}{2}} \cdot 1.8^N$, see [Wat95].
For the number of possible protein structures on the CUB lattice, an asymptotic growth factor of 4.5^N was estimated [MS96].

optimal ones. By choosing low-energy conformations in favor, one tries to reproduce the low-energy portion of the folding landscape as exactly as possible.

A method to favor low-energy conformations is to make use of the common assumption that the distribution of the biopolymer conformations follows the Boltzmann distribution. That is, the probability of a structure x in the ensemble of all possible biopolymer structures in thermodynamic equilibrium is proportional to its *Boltzmann factor*

$$W(x) = e^{-\frac{E(x)}{kT}}, \tag{3.1}$$

in which k is Boltzmann's constant, and T is the temperature of the system. The sum of the Boltzmann factors of all possible conformations is called the partition function Z, as already introduced in Equation (2.7). The Boltzmann factor divided by Z gives the Boltzmann distribution.

In this study, the frequency of choosing an optimum as the start conformation for the sampling was proportional to the Boltzmann weight of the optimum. Conformations being more probable according to the Boltzmann distribution were consequently chosen more often.

As already mentioned, at least one local optimum has to be provided as the start conformation for the sampling. To get a good set of start conformations, optimal and near-optimal structures of the given sequence were predicted. This guaranteed that the lowest part of the energy landscape was covered. The RNA secondary structure prediction was carried out by means of Zuker's algorithm. Suboptimal conformations were generated using the approach of Wuchty et al. For more information about these algorithms, see Section 2.2.1. For proteins, constraint-based protein structure prediction was used as introduced in Section 2.2.2.

The following paragraph presents how the vaguely defined barrier tree BT was realized. This includes both the tree's implementation itself and the implemented operations on the tree.

3.2 The Barrier Tree Data Structure and its Representation

The last section utilized an abstract barrier tree data structure, which will be discussed in detail below. The barrier tree has to meet the demand that the information gained by the sampling is stored in an efficient way to keep down the required amount of memory. Furthermore, the data structure should support several operations on it with moderate time complexity. Miscellaneous representations of a barrier tree are possible. Two of them are discussed in the context of this thesis.

Since a barrier tree is a rooted and weighted graph (compare with Section 2.1), a graph representation suggests itself. The set of all local minima of the landscape, $\mathcal{M}$, corresponds to the vertex set V of the graph. For n optima, a $n \times n$ adjacency matrix with an assigned weight for each edge $(\hat{x}, \hat{y}) \in E$ is required. The entry in row $\hat{x}$ and column $\hat{y}$ is simply the saddle height between the two minima $\hat{x}$ and $\hat{y}$. This adjacency matrix of the graph requires $\mathcal{O}(n^2)$ memory.

From the graph representation, the barrier tree can be calculated. Provided that an ultrametric distance measure is given, the ultrametric tree can easily be reconstructed by an

agglomerative clustering procedure. The *Unweighted Pair Group Method with Arithmetic mean (UPGMA)* [MS57] is an example of these procedures. As stated in Section 2.1, the saddle heights are an ultrametric distance measure. Consequently, the barrier tree, which is constructed from the graph described above with the help of a hierarchical clustering, is the correct tree. The complexity of the hierarchical barrier tree construction algorithm is in the order of the complexity of UPGMA, which is known to be $\mathcal{O}(n^2)$.

In summary, the graph representation of a barrier tree has a space complexity of $\mathcal{O}(n^2)$, and the construction of the tree takes $\mathcal{O}(n^2)$ time.

Another possible representation of the barrier tree is the data structure of a full binary tree. A binary tree is denoted as full, if each node is either a leaf or has degree exactly two. This property is ensured, since the leaves represent the local minima, and the internal nodes represent the saddle heights between them. Because of connecting two minima, each saddle height, and thus each internal node, has two children. Each node of the barrier tree is represented by a single object. To organize the tree, each node has pointer to other nodes. Besides this, each node x has to have a pointer $conf(x)$ to the conformation it represents and to an energy value $E(x)$. The attribute $conf(x)$ is NIL, if x is an internal node. $E(x)$ is the energy of the optima for leaves, or rather it is the saddle height for internal nodes. The attribute $root[BT]$ points to the root of the entire barrier tree BT. The tree is empty if $root[BT] = NIL$. Under the assumption that the cardinality of $\mathcal{M}$ is n, the tree has n leaves. Hence, it has $n-1$ internal nodes and $2n-1$ nodes in total. It follows that this representation has a space complexity of $\mathcal{O}(n)$. During the buildup of the tree, the existing saddle heights are updated, which will be described in detail later on. At this point, it should be mentioned only that this update has a worst-case time complexity of $\mathcal{O}(n)$. If the binary tree is balanced, this bound improves to $\mathcal{O}(\log n)$.

The comparison of the two discussed barrier tree representations points out that a binary tree is evidently the better data structure. The tree representation has also the advantage that the current barrier tree is always available, since it is build up "on-the-fly" during the sampling. This allows printing of the barrier tree during the ongoing sampling process. Furthermore, it can be easily extended by other features. For example, it is possible that, for non-degenerate landscapes, the whole saddle point conformation, instead of the saddle height only, is stored in the binary tree's internal nodes.

The internal use of a binary tree causes a fact which is not really problematic, but quite unaesthetic. Imagine that three optima are mutually accessible by the same saddle height. Since each node in a binary tree has at most two children, a saddle point connecting all three minima cannot be represented. Consequently, the binary tree must have a node that has the same energy value as its child.

The problem is approached by using a clever scheme to represent trees with an arbitrary number of children [CLRS01]: the *left-child, right-sibling representation*, which also uses only $\mathcal{O}(n)$ space for any rooted tree with n nodes. In the scheme, each node x stores the following pointer to other nodes:

1. $parent(x)$ points to the parent of node x,
2. $left-child(x)$ points to the leftmost child of x, and
3. $right-sibling(x)$ points to the sibling of x that is immediately to the right.

The pointer $left-child(x)$ is NIL, if node x has no children. If x is the rightmost child of its parent, then $right-sibling(x) = NIL$. The pointer $parent(x)$ is NIL, if x is the

root of the tree.

The extension of the children list to a doubly-linked list allows the removal of a given node from the tree in constant time. This is achieved by the addition of a left-sibling pointer to each node. Without that pointer, the siblings of the removed node x would have to be traversed to update the right-sibling pointer of x's left sibling.

3.3 Operations on Barrier Trees

The aforementioned sampling algorithm makes use of the following operations which have to be provided by the barrier tree data structure:

- get a random local minimum under the assumption of either a uniform or a Boltzmann distribution
- check whether a given local minimum is already included in the tree
- check whether a given local minimum is part of a shoulder
- insert a new local minimum into the tree and add a given saddle height between the new and a known optima
- update the saddle height between two given optima, if the new height is lower than the current one

It is also useful to have the opportunity to calculate the distance between barrier trees. This permits an impression of how good the barrier tree approximation in comparison to the exact barrier tree is. In the RNA case, the exact barrier tree can be generated with the `barriers` program [FHSW02].

3.3.1 Get a Random Optimum from the Barrier Tree

To return a random optimum from the barrier tree efficiently, a data structure allowing direct access to arbitrary elements in constant time is needed. In this implementation, a vector $Mins$ stores the local minima of the barrier tree. The vector provides random access to its elements by their indices in $\mathcal{O}(1)$ time.

It should be recalled that the distribution of the biopolymer conformations was assumed to follow the Boltzmann distribution. Listing 3.4 shows the algorithm that was employed to get a random Boltzmann distributed optimum from the passed vector $Mins$. The algorithm uses a standard method to randomly choose a value out of several indexed and weighted values.

Listing 3.4: Get a random Boltzmann distributed optimum from the barrier tree

```
GET_BOLTZMANN_DISTRIBUTED_RANDOM_LOCAL_MIN(Mins)
1  create array A[1..length[Mins]]
2  A[1] ← Boltzmann factor of Mins[1]
3  for i ← 2 to length[Mins]
4      do A[i] ← A[i-1] + Boltzmann factor of Mins[i]
5  ▷ choose random number between 0 and A[length[A]]
6  r ← RANDOM(0, A[length[A]])
7  find smallest j: A[j] ≥ r
8  return Mins[j]
```

3.3.2 Check the Existence of an Optimum

A hash table was used in order to check whether a given optimum was already known. Hashing is a very effective and useful technique for the implementation of dictionaries, as the average-case complexity of the basic operations `INSERT`, `SEARCH` and `DELETE` is constant time.

The implementation in the context of this thesis uses a string hash function that was proposed by Daniel J. Bernstein. A string representation of the conformation is used as key. Basically, the function is

$$hash(i) = hash(i-1) * 33 + string[i], \text{ and } hash(0) = 5381.$$

Thus, checking whether a conformation is known, takes $\mathcal{O}(1)$ time on average.

3.3.3 Check Optimum for Being Part of a Shoulder

In terms of Definition (2.2), a local minimum $\hat{x} \in \mathcal{M}$ forms a shoulder $\mathcal{M}(\hat{x})$, if the saddle height between $\hat{x}$ and a $\hat{y} \in \mathcal{M} \setminus \mathcal{M}(\hat{x})$ satisfies $E[\hat{x}, \hat{y}] = E(\hat{x})$. In this case, $\hat{x}$ belongs to the basin $\mathcal{B}([\hat{y}])$ and should therefore not be added as local minimum to the barrier tree.

Likewise, $\hat{x} \in \mathcal{M}$ is, according to Definition (2.2), part of a shoulder $\mathcal{M}(\hat{z})$, if there is a $\hat{y} \in \mathcal{M}(\hat{z})$ with $E(\hat{x}) = E(\hat{y}) = E[\hat{x}, \hat{y}]$. Then, the minima $\hat{x}$ and $\hat{y}$ belong to the same equivalence class and are treated as one minimum.

Since minima that are known to belong to a shoulder are not added to the barrier tree, the latter case can apply only if a known local minimum turns out to be part of a shoulder during the sampling. Therefore, it always has to be checked first, if two local minima are equivalent and if they are consequently members of the same equivalence class. This allows for the removal of the whole class from the barrier tree, if just one class member turns out to be part of the shoulder later on.

Consequently, to test $\hat{x}$ by means of a given saddle height $E[\hat{x}, \hat{y}]$ for being part of a shoulder, it is sufficient to verify if the condition $E[\hat{x}, \hat{y}] = E(\hat{x})$ is satisfied.

3.3.4 Insert New Optimum and Add Saddle Height Between Two Optima

Insert a New Optimum into Barrier Tree

A new optimum is added to the barrier tree by the insertion of a new leaf which references the conformation. Of course, the conformation itself has to be stored at a different place, namely at the vector as mentioned above.

It should be recalled that local minima of degenerate landscapes are collected into equivalence classes. Two minima x and y are member of the same equivalence class $[x]$, if they satisfy the condition $E(\hat{x}) = E(\hat{y}) \wedge E[\hat{x}, \hat{y}] - E(\hat{x}) \leq \varepsilon$ for a given energy threshold ε. Such an equivalence class is represented by just a single leaf in the barrier tree. Still, all optima are saved, which allows that the sampling starts from each of them. The introduction of these equivalence classes helps to keep the barrier tree compact and the landscape smooth. Without them, equivalent minima would blow up the barrier tree, and minima enclosed

by a very low barrier would make the landscape very craggy. Thus, the representation of equivalent minima by just a single leaf is especially necessary if the energy function is highly degenerated. Due to the concept of equivalence classes for local minima, the resulting barrier tree is just a projection of the exact barrier tree. For simplification, it was demanded that the energy threshold ε was equal to 0 for RNA secondary structures and at most 1 for lattice proteins. The restriction avoids that local minima, which are not directly connected to each other by a single saddle point, have to be merged.

Add the Saddle Height Between Two Optima

When a new saddle height sh between the new local minimum a and the known minimum b is found, it does not just provide information about the saddle height between these two optima, but also between a and other local minima. Let s be the new saddle point between a and b, and let t be the highest saddle point which separates b from other local minima with $E(t) < E(s) = sh$. Since there has to be a walk from b to t whose energy never exceeds $E(t)$, there also has to be a walk from t to s whose energy never exceeds $E(s)$. Thus, $t \in \mathcal{V}(s)$ and $\mathcal{V}(t) \subseteq \mathcal{V}(s)$. Then, all local minima in the valley $\mathcal{V}(t)$ can access the minimum $a \in \mathcal{V}(s)$ by the saddle point s with the energy sh as well.

To add the saddle height between a new leaf and a leaf being already part of the barrier tree, the procedure `ADD_SADDLE_HEIGHT` was used as given in Listing 3.5. The procedure is passed a new leaf a, a second leaf b within the tree, and their separating saddle height sh, which will be added to the barrier tree BT.

Listing 3.5: Add the saddle height between two optima

```
ADD_SADDLE_HEIGHT(BT, a, b, sh)
 1  b ← GET_HIGHEST_ANC_BELOW_EBOUND(b, sh)
 2  if parent(b) = NIL
 3     then create node c with energy sh
 4          make a and b children of c
 5          root[BT] ← c
 6     else if e(parent(b)) = sh
 7             then make a child of parent(b)
 8             else create node c with energy sh
 9                  replace b by c in BT
10                  make a and b children of c
```

In line 1, the simple function `GET_HIGHEST_ANC_BELOW_EBOUND` is used to obtain the highest ancestor of b, whose energy is less than the new saddle height sh. The function `GET_HIGHEST_ANC_BELOW_EBOUND` is presented below. The highest ancestor found represents the highest saddle height less than sh which separates b from other local minima. Then, b is set to this highest ancestor. If b has no parent (line 2), it must be the root of the barrier tree. Then, in lines 3–4, a node c representing the new saddle height is created, and a and b become its children. The root of BT is set to c in line 5. If the highest ancestor b is an internal node, two cases can occur. Either the parent of b has the energy sh, then a is added to its children list (lines 6–7). Otherwise, the energy of b's parent is above sh, since b is the highest node below sh. Therefore, in lines 8–10, a new node c representing the saddle height sh is created, the parent of b becomes parent of c, and a and b become c's children.

Instead of being a new leaf, the passed node a can also be the root of a subtree which

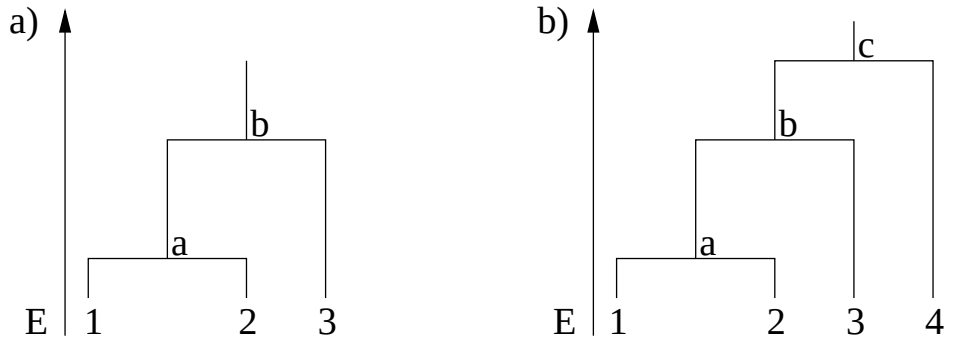


Figure 3.2: Insertion of a new local minimum into a barrier tree. a) Given the barrier tree, a new local minimum labelled with 4 has to be inserted into the barrier tree. The new saddle point c was found between the known minimum 1 and the new minimum 4. The highest ancestor of 1 with an energy below $E(c)$ is the saddle point b, which connects the minima 1 and 3. The node which represents b has no parent and is the root of the barrier tree. b) The new minimum 4 is inserted into the tree. A new node which represents the saddle point c is inserted into the tree. Its children become the minimum 4 and the subtree with the root b. Furthermore, the root of the barrier tree is set to c. The resulting barrier tree is shown in the figure.

is not linked to the barrier tree. Then, the passed saddle height is the energy level that makes the optima within the subtree accessible to b.

Figure 3.2 exemplifies the insertion of a new local minimum into a barrier tree.

Get the Highest Ancestor Below an Energy Bound

The function presented below (Listing 3.6) returns the highest ancestor of the passed node a, whose energy is below the energy bound eb, if it exists. Otherwise, it returns a.

Listing 3.6: Get the highest ancestor below an energy bound

```
GET_HIGHEST_ANC_BELOW_EBOUND(a, eb)
1  anc ← a
2  while parent(anc) ≠ NIL and e(parent(anc)) < eb
3        do anc ← parent(anc)
4  return anc
```

3.3.5 Update the Saddle Height Between Two Optima

In this paragraph, the updating of saddle heights within a barrier tree will be explained. Such an update becomes necessary, when a saddle height between two minima that is lower than the currently known saddle height has been found. The saddle height between two optima is the energy of their *least common acestor (lca)* in the barrier tree. Thus, in order to get the energy that makes two given minima mutually accessible, their least common ancestor has to be determined.

The *lca* of two nodes p and q in a tree is their shared ancestor which is located farthest from the root. Thus, the *lca* is the internal node closest to the nodes p and q that appears in both paths of p and q towards the root of the tree.

Find the Least Common Ancestor of Two Nodes

Derived from its definition, the problem of finding the *lca* of two distinct nodes p and q can be solved with the simple algorithm given in Listing 3.7.

Listing 3.7: Determine the least common ancestor of two nodes

```
DETERMINE_LCA(p, q)
 1  create lists L1, L2
 2  while parent(p) ≠ NIL
 3        p ← parent(p)
 4        add p to L1
 5  while parent(q) ≠ NIL
 6        q ← parent(q)
 7        add q to L2
 8  a ← tail[L1]
 9  b ← tail[L2]
10  lca ← NIL
11  while a ≠ NIL and b ≠ NIL and a = b
12        lca ← a
13        a ← prev[a]
14        b ← prev[b]
15  return lca
```

In lines 2–7, all ancestors of p and q on the path towards the root are collected each into a list. In lines 11–14, the entries of these two lists are compared, starting from the root, until the list entries differ. The last common entry is the *lca* and is returned.

The complexity of the algorithm depends on the height of the tree, since an upward path from each of the two nodes to the root has to be performed. It takes $\mathcal{O}(n)$ time to determine the *lca* in an arbitrary n-node tree, since its height is bounded by the number of nodes. If the tree is approximately balanced, its height is $\mathcal{O}(\log n)$. As barrier trees are, in the majority of cases, balanced to some extent, the complexity of `DETERMINE_LCA` could improve to $\mathcal{O}(\log n)$.

However, Bender and Farach-Colton presented an algorithm that answers lca queries in constant time after only linear preprocessing of the tree [BFC00]. Since their algorithm is effectively implementable, in contrast to other ones presented before, it potentially provides the opportunity to be applied to the sampling approach. However, the algorithm cannot be applied directly as the barrier tree changes dynamically. The algorithm has to be modified in such a way that the preprocessing step does not have to be repeated after each update of the barrier tree. Further analysis would exceed the task of this thesis, but could be the subject of further work in order to improve the runtime of the sampling algorithm.

Update the Barrier Tree

During the sampling process, a saddle height between two minima a and b lower than their currently known saddle height, that is the energy of their *lca* in the barrier tree, can be found. As soon as this happens, the tree has to be updated. Thereby, one of the following three cases can apply:

1. It turns out that the minima a and b are members of the same equivalence class. Therefore, either the leaf representing a or the leaf representing b is removed from the barrier tree, which has to be consolidated after this removal. The consolidation will be explained afterwards.
2. It turns out that a or b belongs to a shoulder. The leaf representing the minimum that belongs to the shoulder is removed from the barrier tree. Subsequently, the tree has to be consolidated.
3. The saddle height between the optima a and b has to be updated in the barrier tree. That is, a lower saddle height between a and b is introduced. Resulting from this, the two subtrees $st1$ and $st2$, which contain the optima a and b respectively and which are below the optima's lca, have to be updated as well. Because the saddle heights have the property to form an ultrametric distance measure (see Section 2.1), the need of the subtree update follows from Equation (2.4): a lower saddle height between the two minima a and b also lowers the saddle height between minimum a and each minimum in subtree $st2$ and between b and each minimum in $st1$. After the subtree update, the tree has to be consolidated.

Consolidate the Barrier Tree After Leaf Removal

The barrier tree update requires a barrier tree consolidation procedure, since the removal of a leaf from the tree might result in an internal node that has just one child. Such a procedure is presented in Listing 3.8. The procedure is passed the parent p of the leaf removed and the barrier tree BT.

Listing 3.8: Consolidate the barrier tree after leaf removal

```
CONSOLIDATE(BT, p)
1  if p has at least 2 children
2     return
3  else if p = root[BT]
4          then root[BT] ← child(p)
5               parent(root[BT]) ← NIL
6          else make child(p) child of parent(p)
7               remove p from parent(p)
8       delete p
```

The procedure `CONSOLIDATE` is returned, if node p has at least two children (lines 1–2). Otherwise, p must have exactly one child, since it is an internal node which had to have at least two children before one of it was removed. If p is the root of the tree, it is deleted and its child becomes the new root of the tree (lines 3–5 and line 8). Otherwise, in lines 6–8, p is spliced out of the tree. That is, the child of node p becomes child of p's parent, and p is deleted from the tree.

Update the Saddle Height Between Two Optima

The update of the saddle height itself, which is the third case mentioned above, is done suchlike that all the leaves of the "smaller" subtree below the lca are relocated into the "larger" subtree below the lca. The "smaller" subtree is named $st1$, and the "larger" subtree is named $st2$. It is assumed that the sampling gives rise to the new saddle height sh

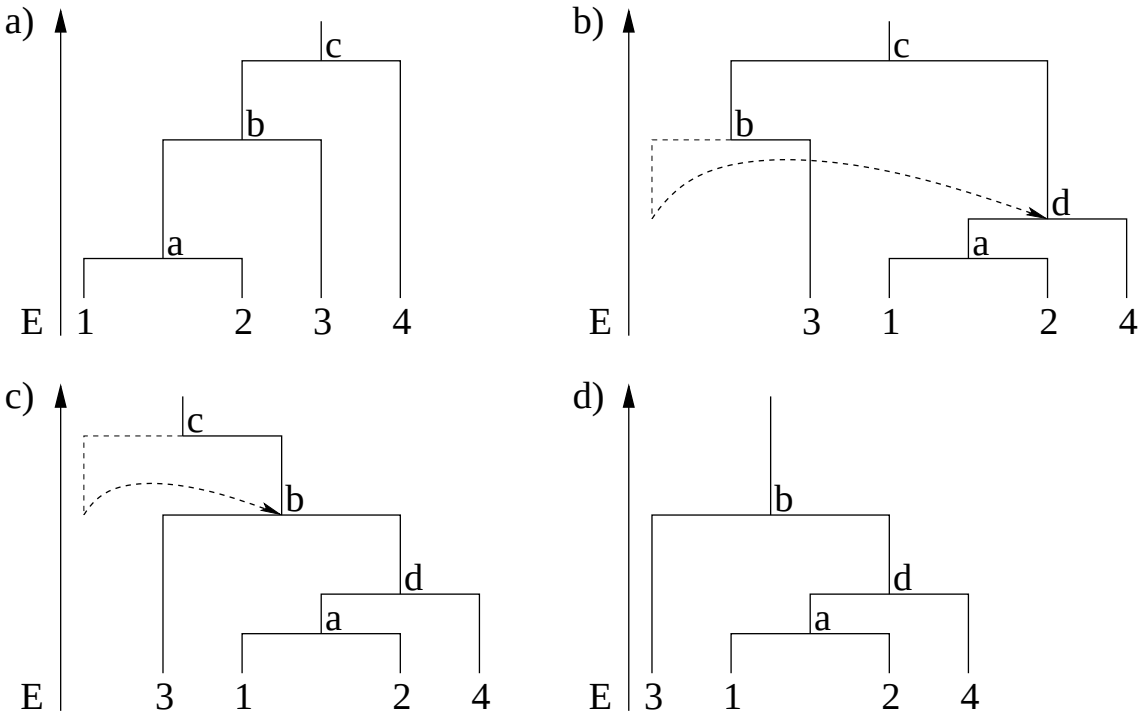


Figure 3.3: Update of a barrier tree. a) Given the barrier tree, the saddle height between the two local minima 1 and 4 has to be updated. The new saddle point between 1 and 4 is d with the saddle height $E(d)$. The current saddle height between the minima 1 and 4 is the energy of their least common ancestor c. Note that although the subtree below the least common ancestor c, which contains just the minimum 4, is definitely the "smaller" subtree below c, it is ascertained that the subtree containing the minima 1, 2, and 3 is the "smaller" one here. Thus, this example can both be concise and illustrate a more complex update process. b) The subtree that contains the minima 1 and 2 has the root a, whose energy $E(a)$ is below $E(d)$. Consequently, this subtree is connected to minimum 4 by the saddle point d. c) The minimum 3 has to be relocated as well, since it is accessible from 1 at the saddle height $E(b)$. d) Because c, the former least common ancestor of 1 and 4, remains with just one child, it has to be spliced out of the tree. This is done by the procedure `CONSOLIDATE`. The barrier tree resulting from the update is shown in the figure.

between the minima a and b. First, the subtree of $st1$ that contains either a or b and whose root has an energy just below the new saddle height sh is determined. Then, this subtree is connected to $st2$ by the saddle height sh. Afterwards, one moves in the remainder of $st1$ towards the lca. On this path, all leaves of $st1$ are relocated into $st2$ until the subtree $st1$ is empty. Finally, the node lca has to be spliced out of the barrier tree, if the root of $st2$ remains its only child. Figure 3.3 exemplifies this algorithm.

The saddle height update procedure for a given barrier tree was implemented as described in Listing 3.9. The procedure is passed the barrier tree BT, the two local minima a and b, their lca and the new saddle height sh between a and b.

Listing 3.9: Update the saddle height in the barrier tree

```
UPDATE_SADDLE_HEIGHT(BT, a, b, lca, sh)
 1  if distance(a,lca) < distance(b,lca)
 2     then swap a and b
 3  subtree1 ← GET_HIGHEST_ANC_BELOW_EBOUND(b,sh)
 4  sub1parent ← parent(subtree1)
 5  remove subtree1 from sub1parent
 6  st2node ← GET_HIGHEST_ANC_BELOW_EBOUND(a, sh)
```

```
 7  ADD_SADDLE_HEIGHT(BT, subtree1, st2node, sh)
 8  while sub1parent ≠ lca
 9        do subtree1 ← sub1parent
10           sub1parent ← parent(subtree1)
11           remove subtree1 from sub1parent
12           st2node ← GET_HIGHEST_ANC_BELOW_EBOUND(st2node, e(subtree1))
13           if (e(parent(st2node)) = e(subtree1))
14              then make subtree1's children to children of parent(st2node)
15                   delete subtree1
16              else replace st2node by subtree1 in BT
17                   make st2node child of subtree1
18  CONSOLIDATE(BT, lca)
```

In the first two lines, the "smaller" subtree $st1$ below the lca is determined. The "smaller" subtree is the subtree wherein the path from the minimum towards the lca is shorter than in the other one. The node denoted with b has to be in $st1$. The distance from the minima towards the lca can be determined by the procedure `DETERMINE_LCA(a, b)` after marginally extending it. In line 3, the algorithm sets the node $subtree1$ to the highest ancestor of b, whose energy is less than sh. Of course, b has an ancestor within the subtree $st1$ below the lca, since $sh < E(lca)$. Then, in lines 4–5, the node $subtree1$ and the subtree below it which contains b is removed from its parent. Subsequently, $subtree1$ is connected by the saddle height sh to the "larger" subtree $st2$ containing a. This is done by means of the procedure `ADD_SADDLE_HEIGHT` (lines 6–7). Afterwards, it is proceeded with the parent of $subtree1$. Lines 8–17 are repeated until the lca is reached. Thus, all remaining leaves from the subtree $st1$ are relocated into the subtree $st2$. In each iteration step, the algorithm moves one tree level closer to the lca (lines 9–10). In line 11, the current node $subtree1$ and the subtree below it is removed from the remainder of the barrier tree. The node $subtree1$ is the saddle point that made the minimum b accessible to the minima within the subtree below the node $subtree1$. Therefore, in lines 12–17, $subtree1$ is inserted into $st2$. If a saddle point with $subtree1$'s energy already exists in $st2$, all of $subtree1$'s children are inserted into $st2$ and just the node $subtree1$ is deleted (lines 13–15). Otherwise, $subtree1$ is inserted into $st2$ as a node (lines 16–17). In line 18, the procedure `CONSOLIDATE` is called, since the node lca might have just one child. Then, it has to be spliced out of the tree.

3.4 Distances Between Barrier Trees

A distance measure on barrier trees makes it possible to interpret the quality of the tree approximation. To compare two such approximations, a reference barrier tree is needed. For certain types of conformation space, this reference can be generated by means of `barriers`. Then, different barrier trees gained by the sampling can be compared with each other in respect of their distance to the reference tree.

The distance measure also gives rise to a new termination condition of the sampling. Instead of specifying the sampling length, the sampling could be terminated as soon as the distance between approximated and reference tree falls below a given threshold.

Packages like PHYLIP [Fel05] already contain methods to compute tree distances. However, they normally allow for differences in tree topology, and the trees should all have the same list of leaves. This is not appropriate for the presented sampling approach, since the differences in the energy barriers, that is the branch lengths, are supposed to be used

for the distance calculation between barrier trees. It is also possible that the compared barrier trees do not represent the same set of local minima. Therefore, the *root mean square deviation* (RMSD) over the saddle heights was used as distance measure.

Once again, low-energy conformations are assessed with a higher weight than conformations with higher energy. The weight assigned to an optima resembles its Boltzmann factor (see Equation (3.1)).

The $RMSD(BT_1, BT_2)$ over two barrier trees BT_1, BT_2, which represent the same set of local minima $\mathcal{M}$, is defined by

$$\sqrt{\frac{1}{N(N-1)} \sum_{x_1 \in \mathcal{M}} \sum_{x_2 \in \mathcal{M}, x_2 \neq x_1} \max\left(W(x_1), W(x_2)\right) \cdot \left(E_{BT_1}[x_1, x_2] - E_{BT_2}[x_1, x_2]\right)^2},$$

where N denotes the cardinality of $\mathcal{M}$, $E_{BT}[x_1, x_2]$ denotes the saddle height between x_1 and x_2 in BT, and $W(x)$ denotes the Boltzmann factor of conformation x.

Assume that BT_1 and BT_2 do not represent the same set of local minima. Then, at least one of the trees contains one or more minima, which are not included in the respective other tree. Let x_{missed} be the minimum of tree BT_1 that is missing in BT_2. Then, the deviation from the saddle heights $E_{BT_1}[x_{missed}, x_n]$ for all $x_n \in \mathcal{M} \setminus \{x_{missed}\}$ cannot be determined, since there are no corresponding saddle heights in BT_2. To resolve this problem, the difference between $E_{BT_1}[x_{missed}, x_n]$ and the height of the highest possible saddle height in BT_2 is calculated. If the maximal saddle height is known, it can be passed as parameter to the function that calculates the RMSD. In the HP-model, this value is 0, which is the energy of the open chain conformation. If, however, the highest possible saddle height is not available, a lower bound for this maximal height will be provided by the maximum of the energy of the two barrier trees' roots.

3.5 Implementation of Energy Landscape Models

The presented sampling approach is generic, this means that it is not dependent on the underlying energy landscape model. Thus, a framework for the study of arbitrary landscapes is needed. The landscape models have to define at least a set of conformations and a neighborhood of the conformations in order to form the conformation space, and an energy function over the conformations. In the context of this thesis, we developed the *Energy Landscape Library (ELL)*[2] that meets these basic requirements [MWB07]. Consequently, this library provides a platform for generic algorithms to investigate energy landscape properties.

3.5.1 General Architecture of the Energy Landscape Library

The core of the ELL is an abstract class `State`, which defines the abstract properties and methods of a state. The term state is equated with the term conformation as used in the last chapter. This class provides the interface between the generic algorithm and its underlying landscape model. A new landscape model is introduced by deriving a subclass from the abstract superclass. This allows for the formulation of the algorithm on an

[2]The ELL is freely available at `http://www.bioinf.uni-freiburg.de/sw/ell/`.

abstract state, which will be specialized afterwards. The generic algorithm on the state, however, does not have to be adapted or reimplemented.

The class `State` provides methods to obtain the fitness of a state, which permits for example evolutionary studies, and to obtain the energy of a state. The energy of a state is its negated fitness. Another basic functionality of the class is the possibility to iterate over the neighbors of a state. This iteration can be done randomly, or with respect to a specific order on the neighbors. To reduce the memory usage for this enumeration, the neighbors are generated on demand. Furthermore, a state can be saved in a compressed representation. This makes it possible to manage a larger number of states, since less memory is used for each of it. The last important method provided by the class is the calculation of distances between two states. The distance is the number of moves that must be applied to one state in order to reach another one.

3.5.2 States for RNA Secondary Structures and Lattice Proteins

The ELL currently implements states for RNA secondary structures (see Section 2.2.1) and for structures of simple lattice proteins (see Section 2.2.2). RNA secondary structures are internally represented in bracket notation and two neighbors can be converted into each other by single moves. The base pair distance defines the distance between two RNA states. The free energy of an RNA secondary structure is calculated as described by Zuker and Stiegler [ZS81] by means of the Vienna RNA package implementation [HFS$^+$94]. The simple lattice protein model supports different lattice types (`SQ`, `CUB`, `FCC`). The sequence alphabet and its associated contact energy function can be arbitrarily assigned. The protein structures are internally represented in relative moves. Hence, neighbors are generated using pivot moves. The distance between two protein structures is the *Hamming distance*[3]. The Hamming distance between two protein states provides a lower bound for the number of relative moves that are necessary to reach one state from the other one. Since only self-avoiding walks are allowed, a direct path[4] connecting two states does not have to exist for lattice protein states, in contrast to RNA states.

Due to the fact that the ELL is highly modular, all available landscape models can be extended in a simple way, and new models can be implemented in a straightforward manner.

3.5.3 Symmetries of Lattice Proteins

As already said in Section 2.2.2, there are symmetrical structures for each lattice protein that result from rotations and reflections of the protein. It was ascertained that every relative move string starts with forwards direction F. Since the first letter of the move string is fixed, several symmetrical structures are already excluded from the conformation space. More precisely, rotations within the $x-y-$layer (that is a rotation around the $z-$axis) and reflections at the $y-z-$layer are forbidden. Consequently, it is not permitted to apply a pivot move to the first position of the relative move string. Such a move would just result in a reflection or rotation of the state instead of generating a new state.

[3] For two strings of equal length, the *Hamming distance* between them is defined by the number of different positions in the strings.

[4] A *direct path* between two conformations is a path where adjacent conformations have a lower distance to the target conformation. For lattice protein states, *minimal refolding paths* instead of direct paths can be defined. These paths allow some indirect steps that result in a larger distance to the target.

The basic lattice protein models, as implemented in the ELL, do not exclude any more symmetrical states beyond the ones described above. The neighbors of a state are generated via point mutations on the relative move string, which can lead to states that are symmetrical to each other.

To exclude the remaining symmetrical structures, the concept of relative move string normalization was developed. As up to now, neighbors are generated by applying a pivot move to a given state. However, two states are considered to be identical, only if their normalized representation is identical. The normalized representation of a lattice protein state is the lexicographically smallest state that is symmetrical to the original state. This lexicographically smallest state is computed by a normalize function. The function changes the letters in the encoding relative move string according to a prior specified order on the relative move alphabet. Consequently, all states that have the same normalized representation are assumed to be members of the same symmetry class, which is represented by its normalized state.

In the following, an example for the `SQ` lattice is given. Assume that the lexicographical order on the relative move alphabet is $FLBR$. Let $FRFLR$ be a SAW encoded in relative moves on the given lattice. The first position of the string does not have to be changed, since F comes lexicographically first. Since the move L comes before the move R at the second string position, all R within the string have to be swapped with L and vice versa. Thus, $FLFRL$ is the resulting, normalized relative move string of the structure being symmetrical to the original one.

A prototype of such a normalize function has been implemented for the cubic lattice. The implementation is based on a replacement scheme for relative moves in symmetrical structures. The scheme ensures that the moves in the relative move strings appear in the lexicographical order $FLDRU$.

Chapter 4

Results and Discussion

In the last chapters, the theoretical background for the study of biopolymer energy landscapes was given. Furthermore, the models as well as the sampling strategy used within this thesis were presented. This chapter will give examples of the sampling and will present the computational results that have been obtained.

The following studies were carried out:

- Two RNA examples were used to assess if the sampling approach is capable of finding the exact barrier tree of the energy landscape. For RNA secondary structures, efficient algorithms exist to enumerate suboptimal structures. Thus, exact barrier trees can be generated as implemented in the program `barriers` [FHSW02]. This provides the possibility to verify the barrier trees obtained by the sampling approach.
- Two lattice protein examples demonstrate the capabilities of the presented approach. In contrast to RNA secondary structures, no efficient algorithm for the enumeration of suboptimal structures below a certain energy level exist for lattice proteins. Hence, no exact barrier trees were available for comparison with the sampled ones. The program `latticeFlooder` [WWH$^+$06] enumerates only neighbored structures below a given energy threshold. Consequently, certain energy barriers cannot be overcome, which is the reason that not all suboptimal structures can be reached.

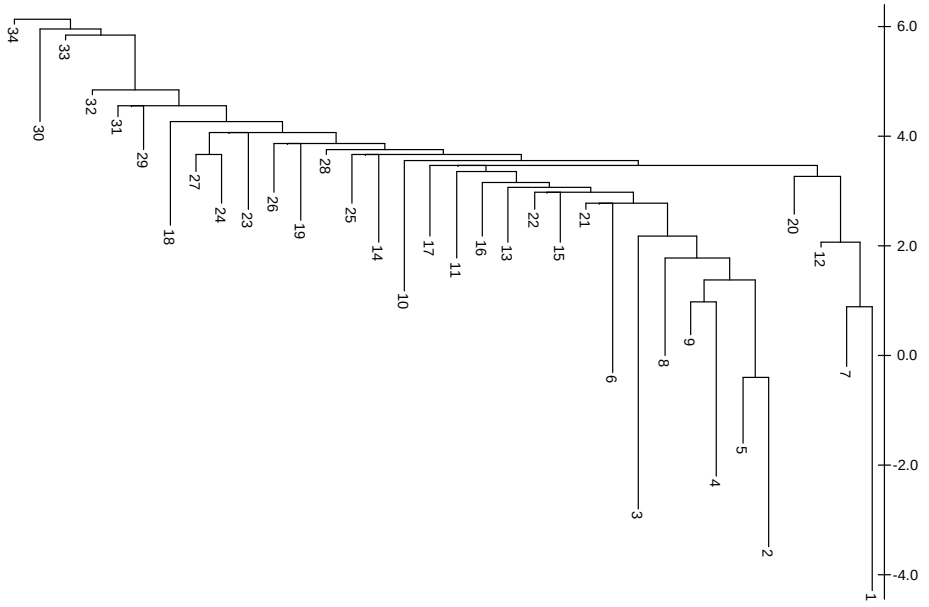
All calculations were performed on AMD Opteron 875 with 2.2 GHz.

To verify the outcomes of the sampling approach, procedures to sort and to print the resulting barrier trees were implemented. The sorting arranges the barrier tree in the same way as the `barriers` program does. It is done in two steps. First, all local minima are sorted by increasing energy. If two conformations have the same energy, they are sorted lexicographically by their string representation. Second, the tree is sorted in the following manner: each node references a conformation which occurs before, according to the conformation order after the first sorting step, the conformation referenced by its left sibling. The barrier tree is sorted breadth-first starting from its root. In addition, printing of the barrier tree in Newick tree format is provided. The barrier tree graphics within this thesis were drawn with the application `NJplot`, which is able to interpret the Newick tree format [PG96].

4.1 Results

4.1.1 Barrier Trees of RNAs

As a first example, an artificially designed RNA molecule of length $n = 20$ with the sequence `CUGCGGCUUUGGCUCUAGCC` was chosen. The sequence has already been presented in previous work [WSSF+04], where it was denoted `xbix`. The conformation space of this molecule consists of 3886 secondary structures. Figure 4.1 shows the exact barrier tree of `xbix`[1]. The tree was calculated by the program `barriers` from a list of all conformations, which was generated using `RNAsubopt`. The barrier tree has 34 local minima. The mfe structure `....((((........))))` has an energy of -4.3 $^{\text{kcal}}/_{\text{mol}}$ and is represented by minimum 1. The denatured, open chain conformation is represented by the local minimum 8.



Local minimum	Secondary structure	Free energy in $^{\text{kcal}}/_{\text{mol}}$
1	`....((((........))))`	-4.3
2	`(((.(((....))).))))..`	-3.5
3	`..........(((....)))`	-2.8
4	`..(((((....)))...)).`	-2.2
5	`....(((....))).....`	-1.6
6	`..((........)).....`	-0.3
7	`....(((..((....)))))`	-0.2
8	`....................`	0.0

Figure 4.1: Barrier tree of the artificially designed RNA sequence `xbix`. The plot shows the barrier tree of the molecule and the table contains the corresponding eight lowest local minima of the energy landscape.

To assess our approach, a sampling of the energy landscape was performed, starting from the mfe conformation `....((((........))))`. The temperature, which is used to adjust

[1]Note that the energies of all RNA secondary structures were calculated with stabilizing energies to unpaired bases adjacent to helices in multi-loops and free ends (dangling ends). Programs like `RNAeval` and `RNAsubopt`, which are part of the `Vienna RNA Package`, thus have to be used with the option `-d1`.

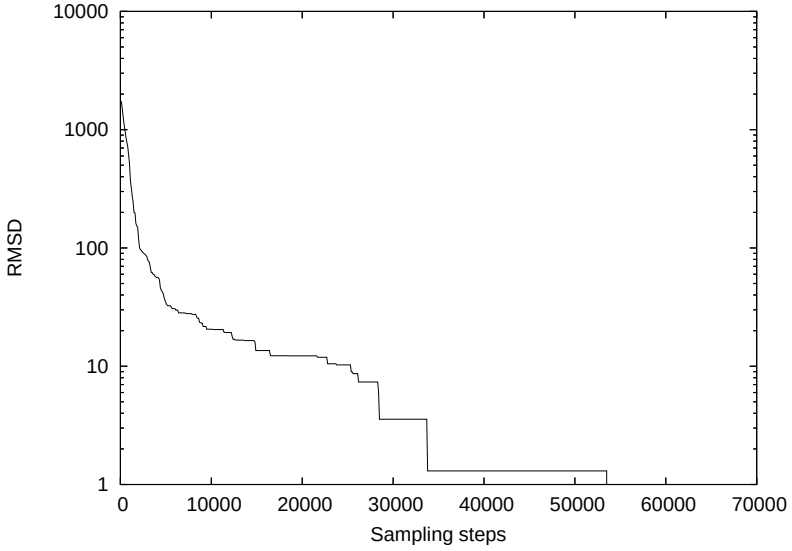


Figure 4.2: Distance between exact and sampled barrier tree of the artificially designed RNA sequence `xbix`. The plot shows the mean RMSD between the two barrier trees versus the number of sampling steps. The mean RMSD is the arithmetic mean of 10 runs, and it is plotted on a logarithmic scale.

the Boltzmann weights of the conformations, was set to 310.15 K. This temperature ensures that all local minima have almost the same probability to be chosen as the start conformation for a single sampling step[2]. Since the complete barrier tree of the energy landscape was of interest, this distribution of the local minima was meaningful. The random walk length of each sampling step was chosen from the uniform distribution on the interval $[1 \ldots 5]$. After every 100 sampling steps, the barrier tree yielded by the sampling was compared with the exact barrier tree of the energy landscape as computed by the program `barriers`. The sampling was terminated as soon as the exact barrier tree has been found. On average over 10 runs, all 34 local minima were found after 1246 sampling steps. The exact barrier tree was found after 25960 sampling steps on average. The mean runtime of a sampling, terminating in the exact barrier tree, was 5 s in our implementation. A plot of the mean RMSD between the exact and the sampled barrier tree versus the number of sampling steps for 10 runs is shown in Figure 4.2. For the calculation of the RMSD, an energy of 23.5 $^{\text{kcal}}/_{\text{mol}}$ was passed as the maximal saddle height. This value is the energy of the highest energy conformation as calculated by `RNAsubopt`. The mean RMSD declines exponentially and reaches the value of 0 after 53600 sampling steps. Figure 4.3 shows the barrier tree of a single sampling run that was stopped after 30000 steps with an RMSD of 13.04 between the sampled and the exact barrier tree. The tree is identical to the exact barrier tree shown in Figure 4.1, except for the position of minimum 23. Therefore, the tree is just an approximation of the exact barrier tree.

The second example is the artificially designed RNA sequence `ACGCGUACGACACGCAACGCAGU` with a length of 23 nucleotides. The conformation space of the RNA molecule with this sequence consists of 6226 secondary structures. The barrier tree in Figure 4.4 shows that the energy landscape of this molecule contains 78 local minima. Minimum 1 corresponds to the mfe conformation `..((((.....))))........` with a free energy of -4.7 $^{\text{kcal}}/_{\text{mol}}$.

[2] A sampling step refers to a single iteration of the sampling algorithm. It consists of a single random walk, the subsequent adaptive walk, and the operations on the barrier tree, if necessary.

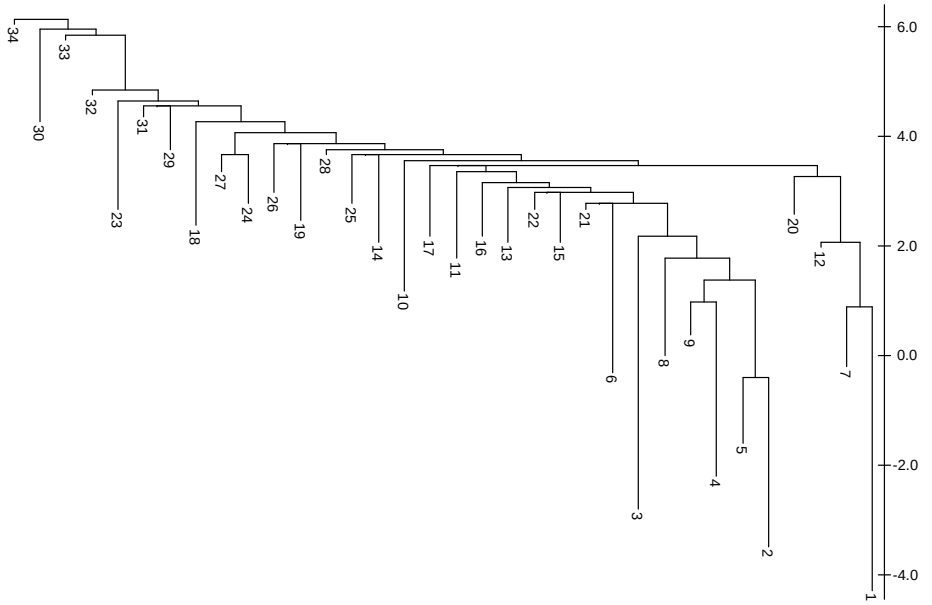


Figure 4.3: Barrier tree approximation of the artificially designed RNA sequence `xbix` after 30000 sampling steps. The energy landscape sampling was stopped after 30000 steps with an RMSD of 13.04 between the sampled and the exact barrier tree. The resulting tree is identical to the exact barrier tree, except for the position of minimum 23.

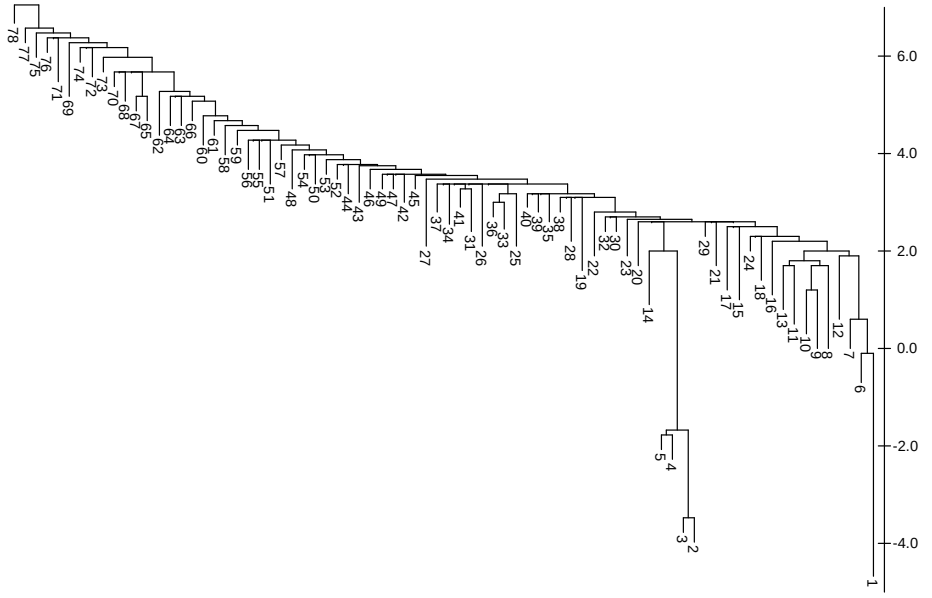


Figure 4.4: Barrier tree of the RNA sequence `ACGCGUACGACACGCAACGCAGU`. The mfe conformation is represented by minimum 1.

Minimum 9 represents the open chain conformation.

The sampling of the energy landscape was performed with the same parameter values as in the previous example. The mfe structure `..((((.....))))........` was provided as start conformation for the calculations. Averaged over 10 runs, all 78 local minima of the energy landscape were found after 2796 sampling steps. The sampling was terminated with the exact barrier tree after 62970 sampling steps on average, which corresponds to a runtime of 15 s in our implementation. Figure 4.5 shows the mean RMSD between exact

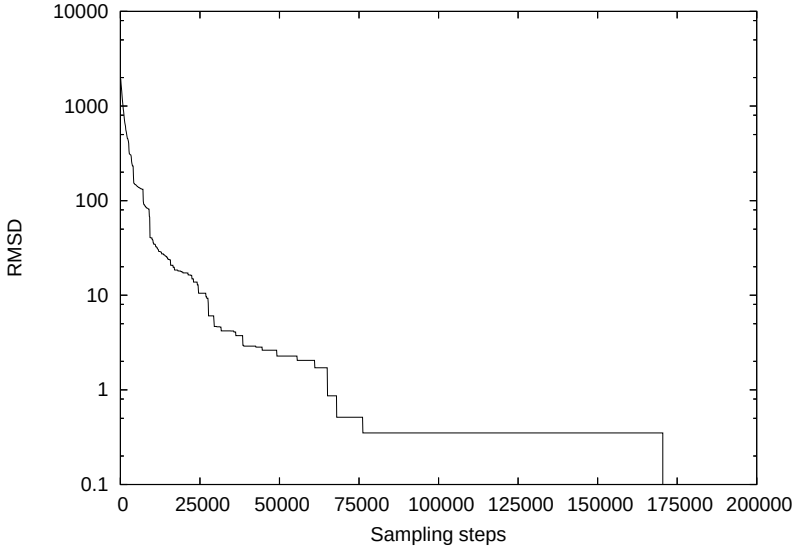


Figure 4.5: Distance between exact and sampled barrier tree of the RNA sequence `ACGCGUACGACACGCAACGCAGU`. The plotted RMSD between the two barrier trees is the arithmetic mean of 10 sampling runs. It is plotted on a logarithmic scale.

and sampled barrier tree plotted versus the number of sampling steps for 10 runs. Again, the energy of the highest energy conformation, which is 25.5 $^{\text{kcal}}/_{\text{mol}}$, was passed as the highest possible barrier height for the RMSD calculation. Similar to the first example, the mean RMSD declines exponentially. A mean RMSD of 0 is reached after 170600 sampling steps.

For the second RNA molecule, another sampling of its energy landscape was performed. This time, shoulders were not excluded from the set of local minima. The resulting barrier tree is shown in Figure 4.6. The barrier tree is similar to the correct one (cf. Figure 4.4), but it contains three additional local minima, namely the minima labelled with 61, 68 and 72. These minima are enclosed by an energy barrier of 0, and each of it is part of a shoulder. In all other examples, minima belonging to a shoulder were excluded from the barrier tree.

4.1.2 Barrier Trees of Lattice Proteins

After discussing the sampling of RNA secondary structure energy landscapes in the last section, the following section will concentrate on lattice protein folding.

In the first example, an HPNX sequence in the three-dimensional cubic lattice was used. The examined lattice protein is a 10-mer with the sequence `HHXHPHNHNP`. The conformation space of this sequence consists of 308981 different SAW structures. When symmetrical structures were excluded, the number of different SAWs on the lattice declined to 39640. Three different parameter settings were used to sample the energy landscape of this protein. In two settings, any SAW structure of the conformation space was allowed to be generated by applying pivot moves during the sampling. In the third setting, normalized structures were allowed only. Such normalized structures reduce the size of the conformation space, since only non-symmetrical structures are considered to be different conformations. Symmetrical structures are collected into symmetry classes. Every sym-

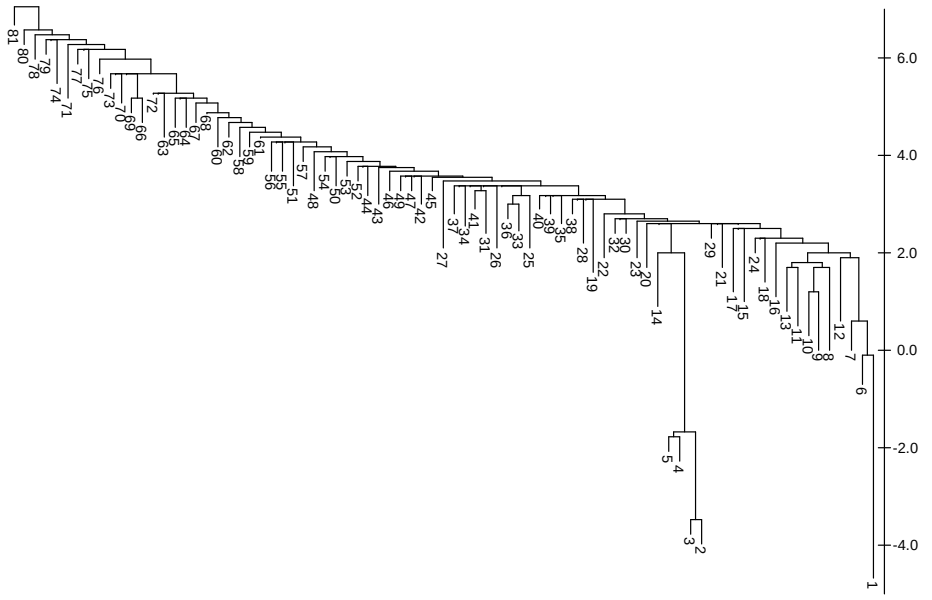


Figure 4.6: Barrier tree of the RNA sequence `ACGCGUACGACACGCAACGCAGU` without exclusion of shoulders. In contrast to the correct barrier tree shown in Figure 4.4, this tree contains three additional minima labelled with 61, 68 and 72. Each of these three minima is part of a shoulder.

metry class is represented by a normalized structure. See Section 3.5 for details about the implementation of the normalize function. Furthermore, in each setting, local minima were collected into equivalence classes. The energy threshold ε for the difference between the minima's energy and their saddle height was set to 0 in setting one and three. It was set to 1 in setting two. All in all, the following three settings were attained:

1. all SAWs on the lattice were allowed as structures, and the threshold ε for collecting local minima into equivalence classes was set to 0,
2. all SAWs on the lattice were allowed as structures, and the threshold ε for collecting local minima into equivalence classes was set to 1,
3. only normalized SAWs on the lattice were allowed as structures, and the threshold ε for collecting local minima into equivalence classes was set to 0.

For the sampling, a temperature of 1 K was chosen. This value favors low-energy conformations. The random walk length of each sampling step was chosen from the uniform distribution on the interval $[1 \dots 10]$. The energy landscape sampling started from an optimal conformation with the relative move sequence `FLLFLFLUU` and an energy of -13. It was terminated after 20 million steps. The runtime of the sampling implementation was about 3 hours for this lattice protein example.

Figure 4.7 shows the barrier trees that were obtained by the sampling of the energy landscape of this lattice protein. From the top to the bottom, the trees were calculated with parameter settings 1, 2 and 3. The upper plot shows the tree resulting from the sampling with parameter setting 1. The tree has 129 leaves, with a single leaf corresponding to an equivalence class of local minima. That is, there are 129 basins, which are associated with 473 different local minima. Additionally, 7404 minima belonging to a shoulder were found, which were not included in the barrier tree. There are 68 different minima with an energy of -13, which is the optimal energy of the examined lattice protein. The basins associated

with the ground states are labeled with 1 to 12. The results were verified with the help of the program `latticeFlooder`, which returned the same 68 ground states with $E = -13$. The plot in the middle shows the barrier tree of the sampling with parameter setting 2. At this, symmetrical structures were not excluded, but minima with the same energy separated by an energy barrier not exceeding 1 were assumed to belong to a common equivalence class. Therefore, the tree shows only 41 different basins. The number of different local minima declined to 193, but the number of minima which belong to a shoulder increased to 7684. Thus, the total number of local minima found, which is 7877, did not change in comparison to the parameter setting discussed before. Once again, 68 ground states with $E = -13$ were found. They belong to basins 1, 2, 3 and 4. The lower plot shows the barrier tree calculated by the sampling which allowed normalized structures only. The tree has 20 basins, which are associated with 66 different local minima. This time, only 9 optimal structures with an energy of -13 were found. These conformations are identical with the ground states as enumerated by `latticeFlooder`. The ground states' basins of attraction are labeled with 1 and 2. The comparison of the barrier trees calculated with parameter settings 1 and 3 leads to the observation that the symmetrical ground states are connected by saddle heights of $E = -12$ and $E = -5$.

To illustrate the convergence of the energy landscape sampling for the sequence `HHXHPHNHNP`, two different features of the resulting barrier trees were chosen. Both the number of local minima found and the average saddle height between each two local minima of the barrier tree versus the number of sampling steps were analyzed. Figure 4.8 displays the curves of the obtained data for the different parameter settings. Each plot shows the arithmetic mean of 5 runs. The general shape of the two different curves is the same for all three parameter settings, and all curves show convergence as well. The sampling over normalized conformations converges much faster to the number of local minima of the landscape than the samplings over all conformations do. In the case of the average saddle height, the curves of the samplings over all conformations decline later and start convergence later than the sampling over normalized conformations does. Furthermore, the curves of the average saddle heights are not smooth, but noisy.

As a second example of lattice protein energy landscapes, the 27-mer HPNX sequence `HHXHPHHHNPHHPHHHHNHPHNHHHNP` in the three-dimensional cubic lattice was chosen. The landscape of this example has just recently been studied using the `latticeFlooder` approach [WWH$^+$06]. To demonstrate the capabilities (and limitations) of the sampling approach, a sampling over the conformation space of this polymer was performed. Constraint-based protein structure prediction (CPSP) was utilized to provide a good set of start conformations for the sampling. The unique ground state with $E = -80$ and four suboptimal conformations on the first energy level with $E = -79$ were chosen as the starting set (see Table 4.1). The energy landscape sampling was stopped after 7 million steps. The temperature was set to 1 K, and the random walk length of each sampling step was chosen from the uniform distribution on the interval $[1 \ldots 10]$. The threshold ε for collecting local minima into equivalence classes was set to 1. With these parameter values, the implementation of the energy landscape sampling took a runtime of several days.

The barrier tree resulting from a sampling, which was performed with the parameter setting as described above, is given in Figure 4.9. The tree shows the 150 lowest local minima. Altogether, 6444934 different local minima and 6395867 associated basins of attraction were found. Since symmetrical structures were not excluded, the tree shows 5 ground states due to rotations and reflections. These states have an energy of -80 and are labeled with 1 to 5 within the tree. Besides this, 20 suboptimal conformations

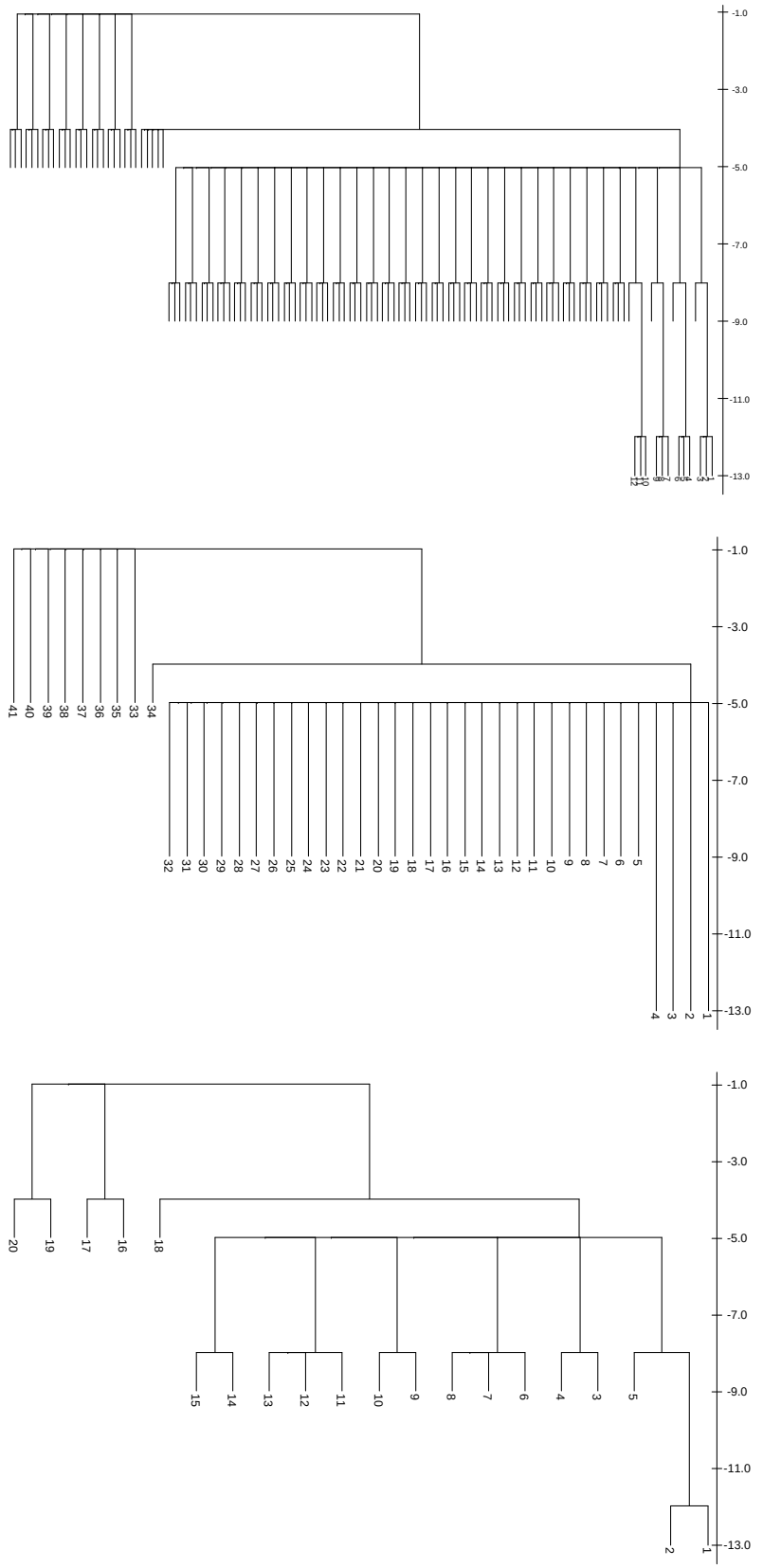


Figure 4.7: Barriers trees of the HPNX-kind lattice protein with sequence `HHXHPHNHNP`. The trees were generated by sampling of the lattice protein energy landscape with parameter settings 1, 2 and 3 (from top to bottom). See text for details.

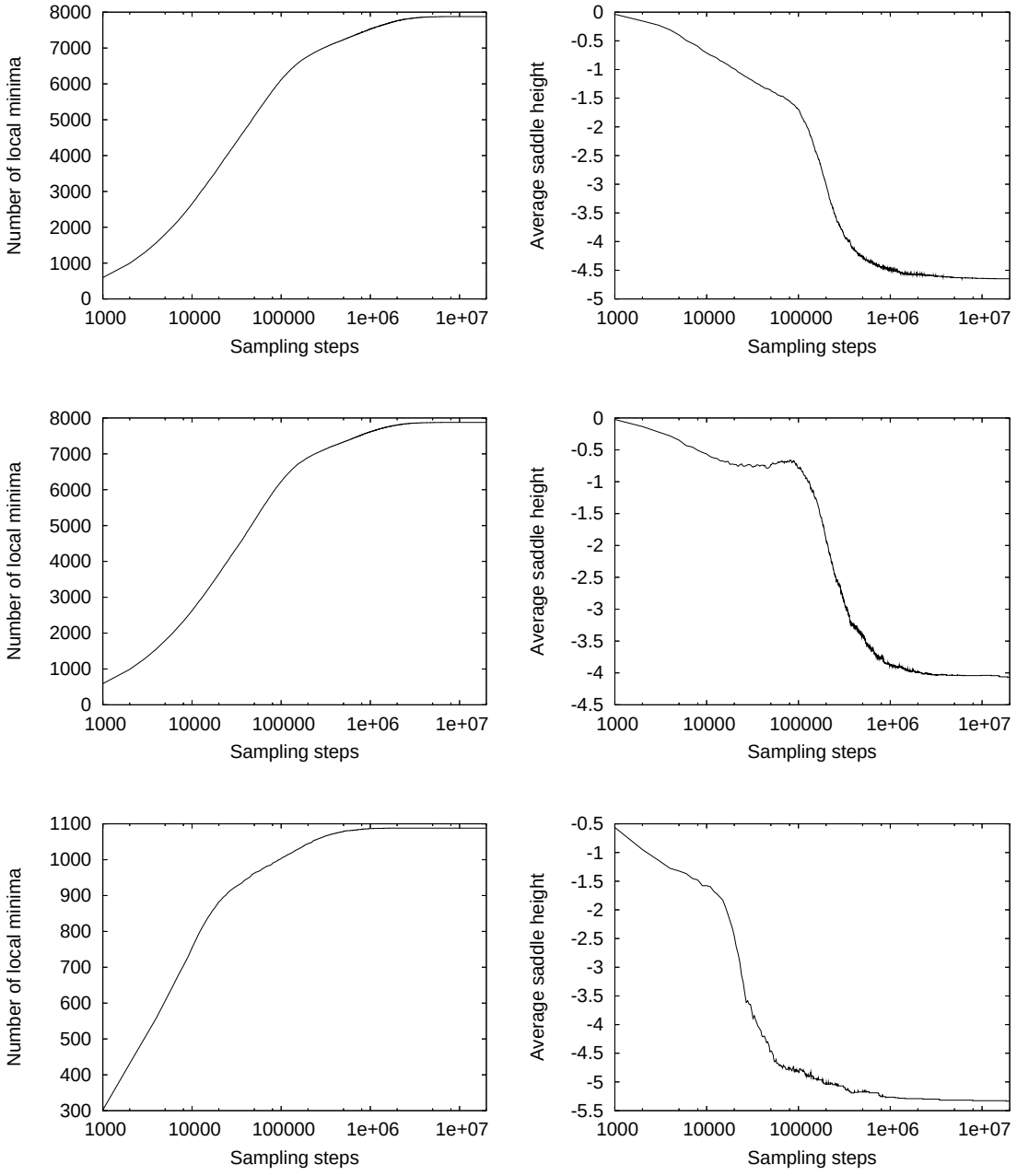


Figure 4.8: Convergence of the energy landscape sampling for the sequence `HHXHPHNHNP`. The two plots in one row belong to samplings with parameter settings 1, 2 and 3 (from top to bottom). The plot on the left shows the number of local minima that have been found during the sampling, including the minima that belong to a shoulder. The plot on the right shows the average saddle height between each two local minima of the resulting barrier tree. Each plot shows the arithmetic mean of 5 samplings per parameter setting. The number of sampling steps is plotted on a logarithmic scale.

Relative moves	Energy
FFLFLFLULLFFLLDDLLRDFDFRRF	-80
FFDFDDULUUFFUUDDLLRDFDDRRD	-79
FFDFDDUFUURFUUDDLLRDFDDRRD	-79
FRRFRFRUDDFFDDRRDDLDFDFLLF	-79
FRRFRFRLRRDFRRUURRDRFRFDDF	-79

Table 4.1: Starting set for the energy landscape sampling of the 27-mer sequence HHXHPHHHNPHHPHHHHHNHPHNHHHNP.

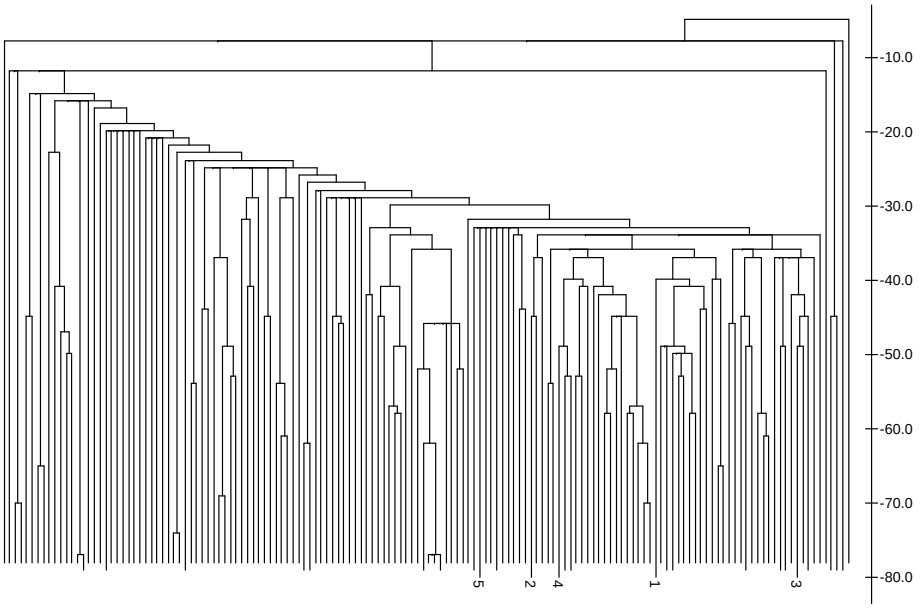


Figure 4.9: Barrier tree of the HPNX-kind lattice protein with sequence HHXHPHHHNPHHPHHHHHNHPHNHHHNP.

with an energy of -79 were found. The studies of Wolfinger et al. [WWH$^+$06] gave a barrier tree with a single minimum with $E = -80$ and 4 minima with $E = -79$. In their results, the optimal and near-optimal conformations were highly connected via saddle heights in an energy range of -40 to -50. The barrier tree found by the sampling approach contains saddle heights at quite high energy levels. The optimal conformations are mutually accessible by energies between -30 to -40. This indicates that the resulting barrier tree appears to be just a rough approximation of the exact one.

4.2 Discussion of Results

In the two RNA examples discussed, the approach for the sampling of energy landscapes yielded the same barrier trees as computed by the program `barriers` from a list of all conformations. With our method, all local minima of the landscape were found after a small number of sampling steps. Consequently, the sampling approach can be used to determine the local minima of biopolymer folding landscapes, at least for short sequences. To compare the barrier tree resulting from the sampling with the exact barrier tree, the RMSD between them was used. At the beginning of the sampling, the RMSD declined rapidly and started to converge to a value of zero. When the sampling was stopped as soon

as the convergence started, it was already obtained a relatively good approximation of the landscape's exact barrier tree. When the sampling was continued, the RMSD reached a value of zero in both examples. An RMSD of zero means that the barrier tree obtained by the sampling agreed with the exact barrier tree.

Accordingly, the two examples showed that the sampling approach is capable of resulting in the correct barrier tree of an energy landscape. The runtime of the sampling implementation was in the range of seconds, which is by all means acceptable.

After verifying the sampling approach with the RNA examples, it was applied to two examples of lattice proteins. In this connection, the exact barrier tree of the energy landscape was not available. Therefore, the convergence behavior of the sampling regarding the number of local minima found and the average saddle height between local minima was analyzed to evaluate the quality of the barrier trees that were obtained.

By putting local minima, which are separated by an energy barrier not exceeding 1, into equivalence classes, the barrier tree resulting from the sampling becomes smaller. More local minima are collected into an equivalence class, which results in fewer classes for the same number of minima. Each leaf in the barrier tree represents such an equivalence class. Therefore, the tree contains less leaves and becomes clearer. However, the resulting barrier tree is just a projection of the exact one.

In the first lattice protein example, one parameter setting excluded symmetrical structures by normalization of relative move strings. The reduced size of the conformation space was reflected in the runtime of the sampling. With this setting, the sampling, and thus the convergence, was faster than with the two other settings. As less conformations and therefore less local minima exist, more walks for every two local minima can be sampled. Consequently, more walks that connect the minima are known after the same number of sampling steps. The number of local minima found versus the number of sampling steps was compared with all three different parameter settings of the first lattice protein example. All samplings showed convergence to a fixed number of local minima, but the sampling over normalized conformations started to converge earlier. The average saddle height in the resulting barrier trees converged to a fixed value as well. Again, the convergence of the average saddle height started earlier for the sampling over normalized conformations. The convergence to a fixed value for all three parameter settings suggests that the sampled barrier trees converge to the correct ones. Of course, it is possible that a certain part of the energy landscape cannot be reached by the sampling by reason of an upper bound for the random walk length that was chosen to be too low. Then, if a part of the landscape is separated from the remainder by a high saddle height, it possibly cannot be reached by the random walk, since the walk is too short to visit a conformation whose energy exceeds this saddle height. To avoid this, the maximal length of the random walk was always chosen higher than the estimated "appropriate value", which was determined following the strategy described in Section 3.1.3. The averaged saddle height plotted versus the sampling steps did not yield a smooth curve, but a curve with some noise. The noise arose due to the removal of shoulder points and the collection of minima into equivalence classes during the sampling. Assume that two minima are already mutually accessible by a low energy. When they turn out to belong to a common equivalence class, the saddle height between them is not used anymore for the calculation of the average saddle height. This results in an increase of the average saddle height, although the barrier tree still converges globally to the exact barrier tree.

When the different barrier trees of the first lattice protein example were compared, it

turned out that certain symmetrical ground states are connected by very low saddle heights of $E = -12$. This can be seen as an artifact of the pivot move set. For small molecules as the investigated 10-mer, only a few intermediate conformations are necessary to turn a given structure into a symmetrical one.

In conclusion, the first lattice protein example indicates that the sampling approach is also capable of returning the correct barrier tree of a lattice protein energy landscape, or at least a good approximation of it.

The sampling results of the second discussed lattice protein example were compared with the results from Wolfinger's study [WWH$^+$06]. With the sampling approach, significantly more local minima were found. Symmetrical structures of optimal and suboptimal conformations were found, and thus a larger region of the energy landscape was covered. The `latticeFlooder` approach selectively enumerates millions of conformations, limited by the size of available RAM. Afterwards, the exact barrier tree of this landscape part is calculated from the enumerated conformations. With this method, generally barrier trees are gained which contain non-connected subtrees. Therefore, with the enumeration method, the need for the sampling of direct or minimal refolding paths between non-connected minima emerges. In contrast, the sampling approach is capable of finding a number of local minima that are in the range of millions. Beyond this, all local minima are connected. An existing drawback of the presented method appears to be that the found saddle heights within the barrier trees are too high. Consequently, they are just an upper bound of the exact saddle heights.

The barrier trees of both lattice protein examples showed that there are several optimal and suboptimal conformations. No unique ground state exists, and there are many conformations with exactly the same energy. This high degree of degeneracy is a common feature of lattice protein energy landscapes. It can be seen as an artifact of the underlying model, which uses a limited alphabet size and fixed bond lengths and angles [WWH$^+$06]. For instance, assume that a single monomer of the lattice protein, which did not contribute to the contact energy of the conformation, was turned. Then, a different conformation arose, but the contact energy of the lattice protein did not change. A study to quantify the degeneracy distribution in three-dimensional HP-models as practical application of CPSP was carried out by Will [Wil05]. At this point, it seems reasonable to ask whether it is correct to model proteins with reduced alphabets as two or four-letter alphabets. Several experimental studies have shown that functional and rapidly folding proteins do not necessarily require the full sequence complexity of naturally occurring proteins (see for example [AKY02]). Fan and Wang investigated reduced alphabet sizes to find the minimum number of amino acid types that is required to encode complex folding proteins. According to them, the lower bound of letters, which is required for a natural protein to encode its structure, is around ten [FW03]. Since the approach presented in this thesis is generic and problem-independent, it could be readily applied to more realistic models with an alphabet that is larger than the four-letter HPNX alphabet. Such extended alphabets would, however, require a method that provides optimal conformations as a starting set for the sampling. Another possibility to circumvent degeneracies would be the choice of more complex lattices like the `FCC` instead of the `CUB` lattice. However, because the degrees of freedom increase in the `FCC` lattice, the size of the conformation space increases as well. Thus, more sampling steps would be required to obtain a barrier tree approximation that would be close to the exact barrier tree.

The comparison of the two approaches selective enumeration and sampling indicates that

they complement each other. The advantages of one method are the disadvantages of the other, and vice versa. The sampling approach is capable of finding a huge number of minima and could therefore be used to roughly characterize the energy landscape. Afterwards, the `latticeFlooder` approach could be used to calculate the exact barrier tree of certain landscape regions by selective enumeration starting from minima which were found by the sampling. Hence, a strategy combining the two algorithms could be the basis for further research programs. Both approaches are problem-independent and applicable to the energy landscape of arbitrary discrete systems. In principle, the barrier trees computed by these methods enable dynamic studies based on landscape theory, even for molecules which have a huge conformation space such as large RNA molecules. The HP and HPNX-model can provide an opportunity to make studies of the dynamic behavior of lattice proteins computationally feasible.

Chapter 5

Conclusion

Barrier trees provide a coarse-grained representation of energy landscapes by organizing local minima and saddle heights in a hierarchical structure. They are therefore a very useful tool for the study of biopolymer folding pathways. This thesis aimed to develop a generic, problem-independent sampling method for the computation of such barrier trees, and to compare the outcome of the sampling with the exact barrier trees obtained by approaches based on enumeration. As a result of this research, a random sampling approach, which allows to compute the exact or approximated barrier tree of the energy landscape, was developed. Using this method, the investigated conformation space of the landscape is not restricted to certain regions. Thus, in comparison to the current approaches for lattice proteins, more local minima were found, and the resulting barrier trees covered a larger region of the energy landscape.

Two examples of RNA molecules were used to compare this method with previous approaches of other groups. Since there was total agreement between the resulting barrier trees, it can be concluded that the sampling approach can be used to compute both all local minima as well as the exact barrier tree of an energy landscape. For RNA molecules and proteins, the size of the conformation space grows exponentially with the sequence length. Since the growth is slower in the RNA case, the exact barrier tree can be sampled for longer sequences than in the protein case.

In order to evaluate the quality of the sampling approach for lattice proteins, the convergence behavior of two features over the sampled barrier tree was analyzed. Both the number of local minima found and the average saddle height between local minima within the barrier tree converged to a fixed value. This convergence strongly indicates that a good approximation of the exact barrier tree was already obtained. It can be assumed that the barrier tree resulting from the sampling converges to the exact barrier tree of the lattice protein energy landscape. The exclusion of symmetrical conformations from the conformation space by normalization of the relative move strings reduces the size of the conformation space. Due to the smaller conformation space, the sampling of the energy landscape starts to converge earlier to the exact barrier tree.

The experiments indicate that the method presented in this thesis is able to significantly find more local minima than former methods based on selective enumeration of certain parts of the lattice protein energy landscape. Consequently, the sampling approach covers the conformation space much better than enumeration approaches. Symmetrical structures of optimal and suboptimal conformations, which did not appear in the outcomes of former

studies, were found. This proves that the sampled barrier trees represent a larger region of the energy landscape. The sampling method has the advantage, that, in contrast to methods based on enumeration, the number of sampled conformations for the barrier tree construction is not basically restricted by the available amount of memory. The size of the resulting barrier tree is, of course, limited, but it is principally possible to reach every conformation of the conformation space by the sampling. An obvious problem of the presented method is, that saddle heights within the resulting barrier trees can be higher than within the exact barrier tree of the energy landscape. The sampling of direct paths between local minima of the barrier tree appears to be a possible way to find better approximations of the saddle heights. Other improvements of the accordance between exact barrier tree and approximated barrier tree could be the subject of further research in this area.

The comparison of different approaches for the exploration of energy landscapes, namely selective enumeration and sampling of conformations, indicates that a strategy which combines the two methods could achieve very promising results. The limitations through the nature of one approach are compensated by the benefits of the other. A conceivable combination could be designed as follows: the sampling is used to roughly characterize the energy landscape of a biomolecule. Subsequently, the lower parts of certain energy landscape regions are selectively investigated by enumeration of conformations starting from optimal and near-optimal conformations found with the help of the sampling. The whole algorithm could be sped up by clever parallelization of sampling and enumeration.

Taking all the different aspects into consideration, it can be said that the sampling approach developed within this thesis appears to be a promising technique for the computation of barrier trees as reduced representation of biopolymer energy landscapes. The barrier trees can be used as the basis for the estimation of basin sizes. Moreover, they are a good starting point for the calculation of biopolymer folding kinetics.

Bibliography

[ABc+97] Richa Agarwala, Serafim Batzoglou, Vlado Dančík, Scott E. Decatur, Sridhar Hannenhalli, Martin Farach, and Steven Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model. *JCB*, 4(3):275–296, 1997.

[AJL+02] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, New York, NY, USA, fourth edition, 2002.

[AKY02] Satoshi Akanuma, Takanori Kigawa, and Shigeyuki Yokoyama. Combinatorial mutagenesis to restrict amino acid usage in an enzyme to a reduced set. *Proc. Natl. Acad. Sci. (USA)*, 99(21):13549–13553, 2002.

[BB97] Erich Bornberg-Bauer. Chain growth algorithms for HP-type lattice proteins. In *Proc. of the First Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 47–55. ACM Press, New York, NY, USA, 1997.

[BFC00] Michael A. Bender and Martin Farach-Colton. The LCA problem revisited. In Gaston H. Gonnet, Daniel Panario, and Alfredo Viola, editors, *LATIN 2000: Theoretical Informatics, 4th Latin American Symposium, Punta del Este, Uruguay, 2000, Proceedings*, volume 1776 of *Lecture Notes in Computer Science*, pages 88–94. Springer, 2000.

[BL98] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *JCB*, 5(1):27–40, 1998.

[BW06] Rolf Backofen and Sebastian Will. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. *Journal of Constraints*, 11(1):5–30, 2006.

[BWBB99] Rolf Backofen, Sebastian Will, and Erich Bornberg-Bauer. Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. *Bioinformatics*, 15(3):234–242, 1999.

[BWC00] Rolf Backofen, Sebastian Will, and Peter Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing (PSB 2000)*, volume 5, pages 92–103, 2000.

[BWF+00] Helen M. Berman, John D. Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein

Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000. The Protein Data Bank home page is `http://www.pdb.org/`, retrieved April 25, 2007.

[CGP+98] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitrou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *JCB*, 5(3):423–65, 1998.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, Cambridge, Massachusetts, USA, 2001.

[Cou02] Jennifer Couzin. Breakthrough of the year: Small RNAs make big splash. *Science*, 298(5602):2296–2297, 2002.

[FDZD98] Adrian R. Ferré-D'Amaré, Kaihong Zhou, and Jennifer A. Doudna. Crystal structure of a hepatitis delta virus ribozyme. *Nature*, 395(6702):567–574, 1998.

[Fel05] Joseph Felsenstein. PHYLIP (phylogeny inference package) version 3.6, 2005. Retrieved January 9, 2007 from `http://evolution.genetics.washington.edu/phylip.html`.

[FFHS00] Christoph Flamm, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. RNA folding at elementary step resolution. *RNA*, 6(3):325–338, 2000.

[FHSW02] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, 216(2):155–173, 2002.

[FW03] Ke Fan and Wei Wang. What is the minimum number of letters required to fold a protein? *JMB*, 328(4):921–926, 2003.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[HFS+94] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte Chemie*, 125(2):167–188, 1994.

[Hig00] Paul G. Higgs. RNA secondary structure: physical and computational aspects. *Q. Rev. Biophys.*, 33(3):199–253, 2000.

[HW97] Arthur L. Horwich and Jonathan S. Weissmann. Deadly conformations – protein misfolding in prion disease. *Cell*, 89(4):499–510, 1997.

[JTZ89] John A. Jaeger, Douglas H. Turner, and Michael Zuker. Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. (USA)*, 86(20):7706–7710, 1989.

[LD89] Kit Fun Lau and Ken A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989.

[LP00] Rune B. Lyngso and Christian N. S. Pedersen. Pseudoknots in RNA secondary structures. In Ron Shamir, Satoru Miyano, Sorin Istrail, Pavel Pevzner, and

Michael Waterman, editors, *Proc. of the 4th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 201–209. ACM Press, New York, NY, USA, 2000.

[LS94] Jon R. Lorsch and Jack W. Szostak. In vitro evolution of new ribozymes with polynucleotide kinase activity. *Nature*, 371(6492):31–36, 1994.

[McC90] John S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6–7):1105–1119, 1990.

[MdWS91] Bernard Manderick, Mark de Weger, and Piet Spiessens. The genetic algorithm and the structure of the fitness landscape. In Rick Belew and Lashon Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150, San Mateo, CA, USA, 1991. Morgan Kaufman.

[MH98] Steven R. Morgan and Paul G. Higgs. Barrier heights between ground states in a model of RNA secondary structure. *J. Phys. A: Math. Gen.*, 31(14):3153–3170, 1998.

[MS57] Charles D. Michener and Robert R. Sokal. A quantitative approach to a problem in classification. *Evolution*, 11:130–162, 1957.

[MS88] Neal Madras and Alan D. Sokal. The pivot algorithm: A highly efficient monte carlo method for the self-avoiding walk. *J. Stat. Phys.*, 50(1–2):109–186, 1988.

[MS96] Neal Madras and Gordon Slade. *The Self-Avoiding Walk*. Probability and Its Applications. Birkhäuser Boston, 1996.

[MSZT99] David H. Mathews, Jeffrey Sabina, Michael Zuker, and Douglas Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol*, 288(5):911–940, 1999.

[MWB07] Martin Mann, Sebastian Will, and Rolf Backofen. The Energy Landscape Library – a platform for generic algorithms. In Sepp Hochreiter and Roland Wagner, editors, *BIRD 2007: Proceedings of Bioinformatics Research and Development*, pages 83–86. Springer, 2007.

[NJ80] Ruth Nussinov and Ann B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. *Proc. Natl. Acad. Sci. (USA)*, 77(11):6309–6313, 1980.

[PG96] Guy Perrière and Manolo Gouy. WWW-query: An on-line retrieval system for biological sequence banks. *Biochimie*, 78(5):364–369, 1996.

[PL95] Britt H. Park and Michael Levitt. The complexity and accuracy of discrete state models of protein structure. *JMB*, 249(2):493–507, 1995.

[RdP02] Nitin Rathore and Juan J. de Pablo. Monte carlo simulations of proteins through a random walk in energy space. *J. Chem. Phys.*, 116(16):7225–7230, 2002.

[RIdP03] Nitin Rathore, Thomas A. Knotts IV, and Juan J. de Pablo. Density of states simulations of proteins. *J. Chem. Phys.*, 118(9):4285–4290, 2003.

[RIdP06] Nitin Rathore, Thomas A. Knotts IV, and Juan J. de Pablo. Confinement effects on the thermodynamics of protein folding: Monte carlo simulations. *Biophys. J.*, 90(5):1767–1773, 2006.

[RTV86] Rammal Rammal, Gérard Toulouse, and Miguel Angel Virasoro. Ultrametricity for physicists. *Rev. Modern Phys.*, 58(3):765–788, 1986.

[SS04] Peter Schuster and Peter F. Stadler. Discrete models of biopolymers. In M. James C. Crabbe and Andrzej Konopka, editors, *Handbook of Computational Chemistry and Biology*, pages 187–221. Marcel Dekker, New York, NY, USA, 2004.

[Sta02] Peter F. Stadler. Fitness landscapes. In Michael Lässig and Angelo Valleriani, editors, *Biological Evolution and Statistical Physics*, pages 187–207, Berlin, Germany, 2002. Springer-Verlag.

[STD+03] Guang Song, Shawna Thomas, Ken A. Dill, J. Martin Scholtz, and Nancy M. Amato. A path planning-based study of protein folding with a case study of hairpin formation in protein G and L. In *Proceedings of the Pacific Symposium on Biocomputing 2003 (PSB 2003)*, pages 240–251, 2003.

[tDPD92] Edwin ten Dam, Kees Pleij, and David Draper. Structural and functional aspects of RNA pseudoknots. *Biochemistry*, 31(47):11665–11676, 1992.

[Wat95] Michael S. Waterman. *Introduction to computational biology: maps, sequences and genomes*. Chapman & Hall, London, UK, 1995.

[WFHS99] Stefan Wuchty, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–65, 1999.

[Wil05] Sebastian Will. *Exact, Constraint-Based Structure Prediction in Simple Protein Models*. PhD thesis, Friedrich-Schiller-Universität Jena, Germany, April 2005.

[WL01a] Fugao Wang and David P. Landau. Determining the density of states for classical statistical models: a random walk algorithm to produce a flat histogram. *Phys. Rev. E*, 64(5):056101–0561016, 2001.

[WL01b] Fugao Wang and David P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Phys. Rev. Lett.*, 86(10):2050–2053, 2001.

[Wol04] Michael T. Wolfinger. *Energy Landscapes of Biopolymers*. PhD thesis, Universität Wien, Austria, October 2004.

[Wri32] Sewall Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth International Congress of Genetics*, volume 1, pages 356–366, 1932.

[WS78] Michael S. Waterman and Temple F. Smith. RNA secondary structure: a complete mathematical analysis. *Math. Biosci.*, 42(3–4):257–266, 1978.

[WSSF+04] Michael T. Wolfinger, W. Andreas Svrcek-Seiler, Christoph Flamm, Ivo L. Hofacker, and Peter F. Stadler. Efficient computation of RNA folding dynamics. *J. Phys. A: Math. Gen.*, 37(17):4731–4741, 2004.

[WTK+94] Amy E. Walter, Douglas H. Turner, James Kim, Mathew H. Lyttle, Peter Müller, David H. Mathews, and Michael Zuker. Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl. Acad. Sci. (USA)*, 91(20):9218–9222, 1994.

[WWH+06] Michael T. Wolfinger, Sebastian Will, Ivo L. Hofacker, Rolf Backofen, and Peter F. Stadler. Exploring the lower part of discrete polymer model energy landscapes. *Europhys. Lett.*, 74(4):726–732, 2006.

[ZS81] Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.

List of Figures

List of Tables

List of Listings

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, den .....

Unterschrift