Path Abstractions in RNA Landscapes

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
ALBERT-LUDWIGS UNIVERSITY OF FREIBURG
MAY 2009

Done by: Sergiy Bogomolov
Born on: 19.12.1986

Supervisors: Prof. Dr. Rolf Backofen
Prof. Dr. Andreas Podelski
Martin Mann

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Freiburg, den 27. Mai 2009

Zusammenfassung

RNAs nehmen in Zellen an verschiedenen Prozessen teil. Man kann Energielandschaften benutzen um den RNA Strukturraum zu charakterisieren. Deshalb kann man mit diesen Energielandschaften die Prozesse, bei denen die verschiedenen RNAs beteiligt sind, besser verstehen. Es ist wichtig die Energiebarriere in RNA Landschaften in vielen praktischen Problemen abzuschätzen (zum Beispiel bei der kinetischen RNA Faltung (Geis *et al.*, 2008) oder bei der Suche nach bistabilen RNA Molekülen (Flamm *et al.*, 2001)). Zu diesem Problem wurden einige Ansätze entwickelt. Man sollte diese Ansätze in zwei Punkten verbessern: verringerte Zeitkomplexität und gleichzeitig die Präzision von Abschätzungen erhöhen. Diese Masterarbeit hat als Ziel die Untersuchung von den Lösungen zu den oben erwähnten Problem. Wir wenden "shape abstraction" auf das Problem der Barriereabschätzung an. In der Masterarbeit wurden einige, auf dieser Abstraktion basierende, präzisere Algorithmen entwickelt und mit den schon existierenden Ansätzen verglichen.

Abstract

RNAs take part in diverse processes in cells. Energy landscapes can be used to characterize the structural space of an RNA and thus can help us to better understand the processes in which RNAs are involved. The task of estimating energy barriers in RNA landscapes is important in many practical problems such that kinetic RNA folding (Geis *et al.*, 2008) and search for bistable RNA molecules (Flamm *et al.*, 2001). A few approaches has been developed to solve this problem. They need to be improved in two ways: improve time complexity and, at the same time, improve the accuracy of estimations. This master thesis has a task of investigating possible solutions to above-mentioned problem. We apply "shape abstraction" to the barrier height estimation problem. In the master thesis a number of precise algorithms based on this abstraction have been developed and compared to already existing ones.

Acknowledgements

I would like to take the opportunity to thank the people who have supported me through my Master experience. First of all I would like to say thanks to Prof. Dr. Andreas Podelski and Prof. Dr. Rolf Backofen, who gave me an interesting topic and helped me during the work on the Master thesis.

Second, I would like to thank Martin Mann for answering lots of my questions and helping me with new ideas and algorithms.

Finally, I would like to thank my parents and my beloved girl-friend Ievgeniia. Without your support and love this work would not be possible.

Contents

Chapter 1

Introduction

1.1 Motivation

Over the last 10 years it became evident that RNA plays a central role within living cells and actively performs a lot of tasks in many different biological contexts. These functions are often related to the three-dimentional structure of the molecules. But the basic properties of the energy landscape of an RNA molecule can be characterized using RNA secondary structures (Flamm *et al.*, 2000). RNA energy landscapes can help us to understand the folding mechanisms of RNAs.

In (Geis *et al.*, 2008) a heuristic approach to kinetic RNA folding that constructs secondary structures by stepwise combination of building blocks is presented. These blocks correspond to sub-sequences and their thermodynamically optimal structures. Optimal structures are calculated using dynamic programming approach. Morgan-Higgs heuristic and a barrier tree based heuristic are used to model folding trajectories. In the paper it is emphasized that the performance of the whole approach crucially depends on approximating saddle heights and therefore further improvements to the Morgan-Higgs heuristic as well as alternative approaches should be investigated.

It is known that non-native conformations can have energies comparable to the ground state and they can be separated from the native state by very high energy barriers. Because of that it is needed a lot of energy to reach the native state. The RNA folding process can be slowed down when the structure is misfolded. Alternative conformations of the same RNA can determine completely different functions (Baumstark *et al.*, 1997). *Molecular* switches that regulate and control a number of biological processes are based on the capability of RNA molecules to form multiple (meta)-stable conformations with different functions (Perrotta & Been, 1998; Zamora *et al.*, 1995).

Flamm *et al.* (2001) have shown that bistable, and more generally, multistable RNA molecules with a variety of additional properties can be found rather easily. A computational method that allows the design of RNA sequences that fold into prescribed alternative conformations is presented. It is crucial for this method to efficiently and precise approximate energy barriers. This follows from the fact that the energy barriers separating local minima are the most important factor influencing the folding kinetics of an RNA (Flamm *et al.*, 2000). Thus we can see that finding approximations for barriers heights is an important task in many areas of research.

1.2 Contribution

In the master thesis new algorithms which use shape abstraction for estimating barrier heights have been developed. These algorithms have been experimentally compared to already existing approaches to the problem. Using the developed algorithms one get more precise estimations and nevertheless the algorithms are feasible for long RNA sequences.

1.3 Related work

Uejima & Hagiya (2004) improve Morgan-Higgs Heuristic (Morgan & Higgs, 1998) by using base pair incompatibility graph and introducing ordering of base pairs under consideration. An improved version of Morgan-Higgs Heuristic was also developed by Geis *et al.* (2008). One has added two parameters that affect the frequency of building and the treatment of conflict groups. First parameter defines the maximum length of partial paths under consideration and second parameter determines whether to recalculate the conflict group after certain number of base pairs have been added to the current structure. Geis *et al.* (2008) also proposes two further modifications to the heuristic that the user can choose. The first allows the folding of partial trajectories in the case that the entire trajectory between structures crosses an energy barrier that is too high. Furthermore, one may make base pair transitions more realistic by only allowing one stack of less than 3 base pairs at a time. Finally, Flamm *et al.* (2001) uses breadth-first-search to find approximations of barriers. On each step of BFS several best structures are saved. We iterate until we reach the target structure. This method is considered in more detail in chapter 4.2.

1.4 Overview

In Chapter 2 some preliminaries and definitions are given. Chapter 3 presents exact methods for calculating barriers. In Chapter 4 some known and new heuristic methods are considered. In Chapter 5 the experimental results are discussed. Finally in Chapter 6 the results are summarized and the outlook of possible further research in this area is given.

Chapter 2

Preliminaries and Fundamental Concepts

2.1 RNA

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 protude 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.

Definition 2.1.1 (RNA Structure)**.** Let $s \in \{A, C, G, U\}^*$ be a sequence. Then, an *RNA structure* over s is a set P of pairs

$$P = \{(i, j) \mid i < j \wedge s_i, s_j \text{ form a Watson-Crick or a non-standard base pair (G-U)}\}.$$

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

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

A structure with the second property is called non-crossing and does not contain pseudo-knots. Pseudo-knots play an important role in many natural RNAs (Ten Dam *et al.*, 1992). Since we can efficiently compute energy only of pseudo-knot free structures (Zuker & Stiegler, 1981), we will consider only pseudo-knot free structures in the remainder of the thesis. In Figure 2.1 (the picture is taken from (Kochniss, 2008)) a detailed picture of the RNA sequence AGUC is presented.

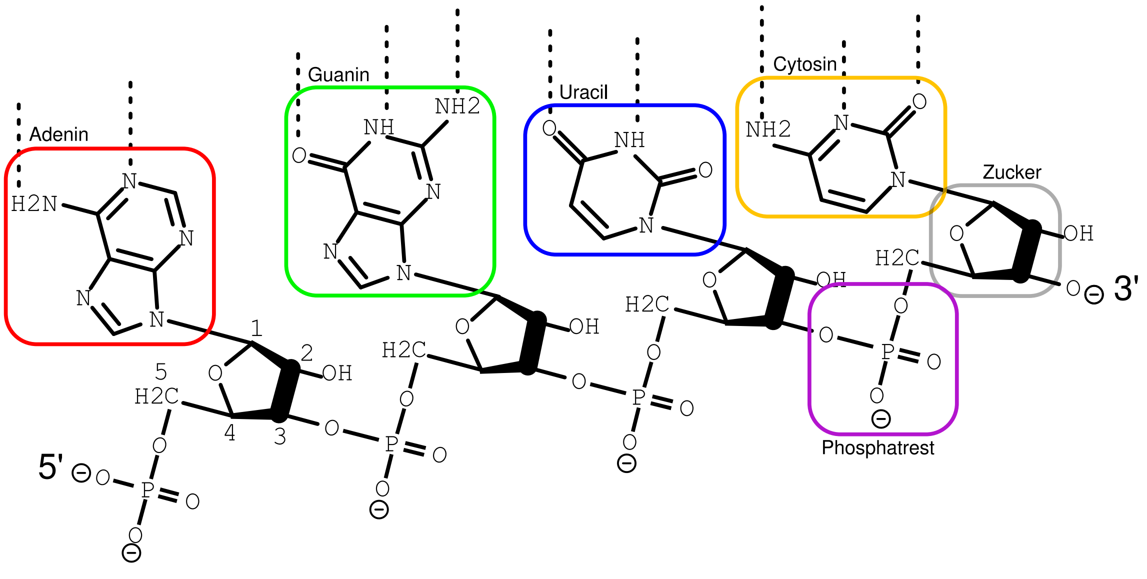

Figure 2.1: Picture of the RNA sequence AGUC

In order to define abstract shapes of RNA in Section 2.4 we will need the following definitions.

Definition 2.1.2 (RNA Structural Elements)**.** Let S be a fixed sequence. Further, let P be an RNA structure for S.

- a base pair $(i, j) \in P$ closes a *hairpin loop* if $\forall i < i' \leq j' < j : (i', j') \notin P$.
- a base pair $(i, j) \in P$ closes a *stacking* if $(i + 1, j - 1) \in P$.
- two base pairs $(i, j) \in P$ and $(i', j') \in P$ form an *internal loop* (i, j, i', j') if
  - $i < i' < j' < j$
  - $(i' - i) + (j - j') > 2$ (no stack)
  - there is no base pair (k, l) between (i, j) and (i', j').
- An internal loop is called *left* (respectively *right*) *bulge*, if $j = j' + 1$ (respectively $i' = i + 1$).
- A *k-multiloop* consists of multiple base pairs $(i_1, j_1), \ldots, (i_k, j_k) \in P$ with a closing base pair $(j_0, i_{k+1}) \in P$ with the property that
  - $\forall 0 \leq l \leq k : (j_l < i_{l+1})$

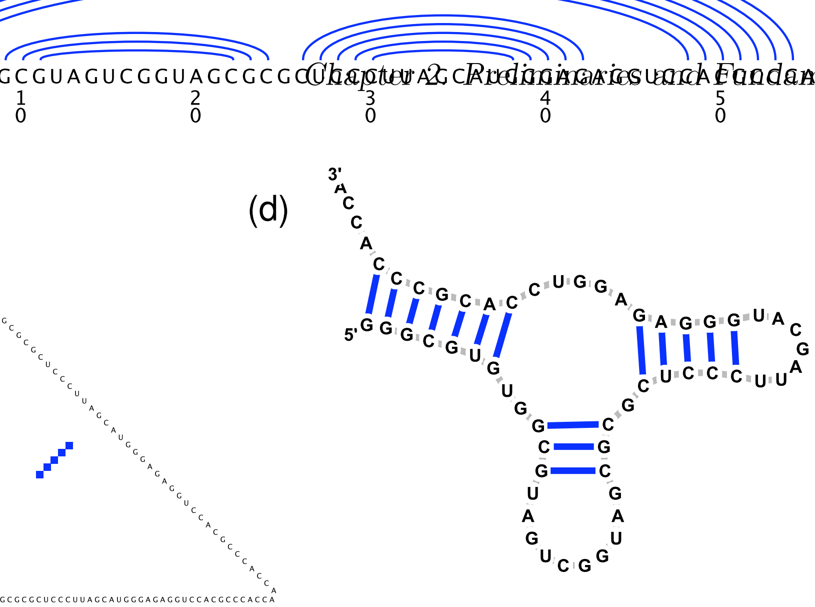

Figure 2.2: RNA secondary structure plot

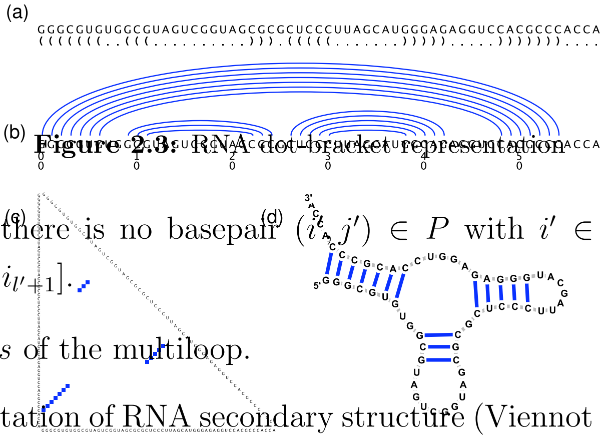

Figure 2.3: RNA dot-bracket representation

  - $\forall 0 \leq l, l' \leq k$ is true that there is no basepair $(i', j') \in P$ with $i' \in [j_l, \ldots, i_{l+1}]$ and $j' \in [j_{l'}, \ldots, i_{l'+1}]$.

- $(i_1, j_1), \ldots, (i_k, j_k)$ close the *helices* of the multiloop.

Definition 2.1.3 (Dot-bracket representation of RNA secondary structure (Viennot & De Chaumont, 1983))**.** For $\Sigma = \{(,),.\}$ and $w \in \Sigma^*$ let $|w|_x$ for $x \in \Sigma$ denote the number of occurrences of symbol x in w. Then a word $w \in \Sigma^n$ is a secondary structure of size n if w satisfies the three following conditions:

1. For every factorization $w = u \cdot v$, $|u|_(\geq |u|_)$.

2. $|w|_(= |w|_)$.

3. w has no factor ().

In Figures 2.2 and 2.3 (the pictures are taken from (Kochniss, 2008)) a RNA secondary structure plot and RNA dot-plot representation respectively are shown.

2.2 Energy Landscape

In order to characterize the space of possible RNA structures will use the notion of energy landscape. Energy landscape is the particular case of fitness landscape which was introduced in (Wright, 1932). The idea of fitness landscape can be used in different areas, e.g. in combinatorial optimization problems.

Definition 2.2.1 (Energy landscape)**.** An energy landscape can be described formally by the following three parts:

1. A set X of structures

2. an operator $N : X \to \mathcal{P}(X)$, which defines the neighborhood of a conformation $x \in X$, and

3. an energy function $E : X \to \mathbb{R}$.

Definition 2.2.2 (Structural space)**.** The structural space $\mathcal{X}$ is formed by the structural set X in combination with the neighborhood operator N. It can be distinguished between discrete landscapes, which have a finite structural space, and continious landscapes (e.g. off-lattice protein models (Stillinger & Head-Gordon, 1995)). In the following we will discuss only discrete landscapes. We also will use RNA conformation and structure as synonyms.

Definition 2.2.3 (Move set)**.** 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 (Stadler, 2002). The move sets we use here assign to each conformation $x \in X$ a set $N(x)$ of accessible neighboors. $N(x)$ 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 $(x, y) \in \mathfrak{N} \Leftrightarrow y \in N(x)$. In the following we will consider the *single move set* which allows deletion or addition of one bond.

Definition 2.2.4 (Structure energy)**.** The energy of an RNA structure is assumed to be equal to the sum of contributions of all structural elements

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

where $E^P_{i,j}$ is the energy contribution of the structural element defined by the base pair (i, j) (see Definition 2.1.2).

Definition 2.2.5 (Local minimum)**.** A conformation $\hat{x}$ is called a *local minimum*, if

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

We write "$\leq$" in the definition because some structures can have in general the same energy (We call energy landscape where structures with the same energy are allowed *degenerate* energy landscapes. In this work we will only consider *degenerate* energy landscapes).

Definition 2.2.6 (Global minimum)**.** A conformation $\hat{x}$ is called a *global minimum*, if

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

Obviously each global minimum is also a local minimum.

Definition 2.2.7 (Walk)**.** A *walk* between the conformations x and y is the list of conformations

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

Definition 2.2.8 (Random walk)**.** *Random walk* denotes an arbitrary, randomly chosen walk between two conformations.

Definition 2.2.9 (Adaptive walk)**.** A walk is called an *adaptive walk*, if for the list of the conformations $x_1, \ldots, x_k$ the following condition holds:

$$\forall 1 \leq i < k : E(x_{i+1}) \leq E(x_i) \wedge \nexists y \in N(x_k) : E(y) \leq E(x_k).$$

Definition 2.2.10 (Gradient walk)**.** A walk is called a *gradient walk*, if for the list of the conformations $x_1, \ldots, x_k$ the following condition holds:

$$\forall 1 \leq i < k : E(x_{i+1}) \leq E(x_i) \wedge x_{i+1} = \operatorname*{arg\,min}_{x \in N(x_i)} E(x) \wedge \nexists y \in N(x_k) : E(y) \leq E(x_k).$$

That is, in each step of the gradient walk, the neighbour with the minimal energy has to be chosen.

Definition 2.2.11 (Length of walk)**.** *Length* of a walk $\mathbf{w}$ is the number of moves in the walk $\mathbf{w}$ (denoted as $L(\mathbf{w})$).

Definition 2.2.12 (Direct walk)**.** A *direct* walk is the shortest path in energy landscape, i.e. a walk $\hat{\mathbf{w}}$ between $\hat{x}$ and $\hat{y}$ is called *direct* if

$$L(\hat{\mathbf{w}}) = \min\{L(\mathbf{w}) \mid \mathbf{w} : \text{ walk between } \hat{x} \text{ and } \hat{y}\}.$$

Definition 2.2.13 (Direct walk in case of RNA)**.** In the case of RNAs, a walk between two conformations S_1 and S_2 is called *direct*, if it only considers direct routes, that is walks that only change base pairs in the symmetric difference $S_1 \triangle S_2$ of S_1 and S_2.

Definition 2.2.14 (Mutually accessible conformations)**.** Two conformations x and y in X are called mutually accessible at the level η, written

$$x \hookleftarrow \eta \hookrightarrow y,$$

if there is a walk $\mathbf{w}$ in $\mathcal{X}$ from x to y, such that $\forall z \in \mathbf{w} : E(z) \leq \eta$ (Flamm *et al.*, 2002).

Definition 2.2.15 (Barrier height)**.** The *barrier height* $E[\hat{x}, \hat{y}]$ between $\hat{x}$ and $\hat{y}$ is the minimum height which makes them accessible from each other, that is

$$E[\hat{x}, \hat{y}] = \min\{\max[E(s) \mid s \in w] \mid w : \text{walk from } \hat{x} \text{ to } \hat{y}\} = \min\{\eta \mid \hat{x} \hookleftarrow \eta \hookrightarrow \hat{y}\}$$

A point $s \in X$ satisfying this condition is called a *barrier* between $\hat{x}$ and $\hat{y}$.

The local minima and the barriers between them can be represented in a hierarchical structure. This hierarchical structure is called the *barrier tree* of the energy landscape. Formally barrier tree is defined below.

Definition 2.2.16 (Barrier tree)**.** The *barrier tree* is a rooted graph $G(V, E)$. The vertex set V contains the local minima of the landscape and the barriers connecting them. Each vertex has an associated energy value, which is the energy of the local minimum and the barrier, respectively. The leaves of the tree are the local minima, and the internal nodes represent the barriers.

In Figure 2.4 a barrier tree for the sequence subROSE is presented.

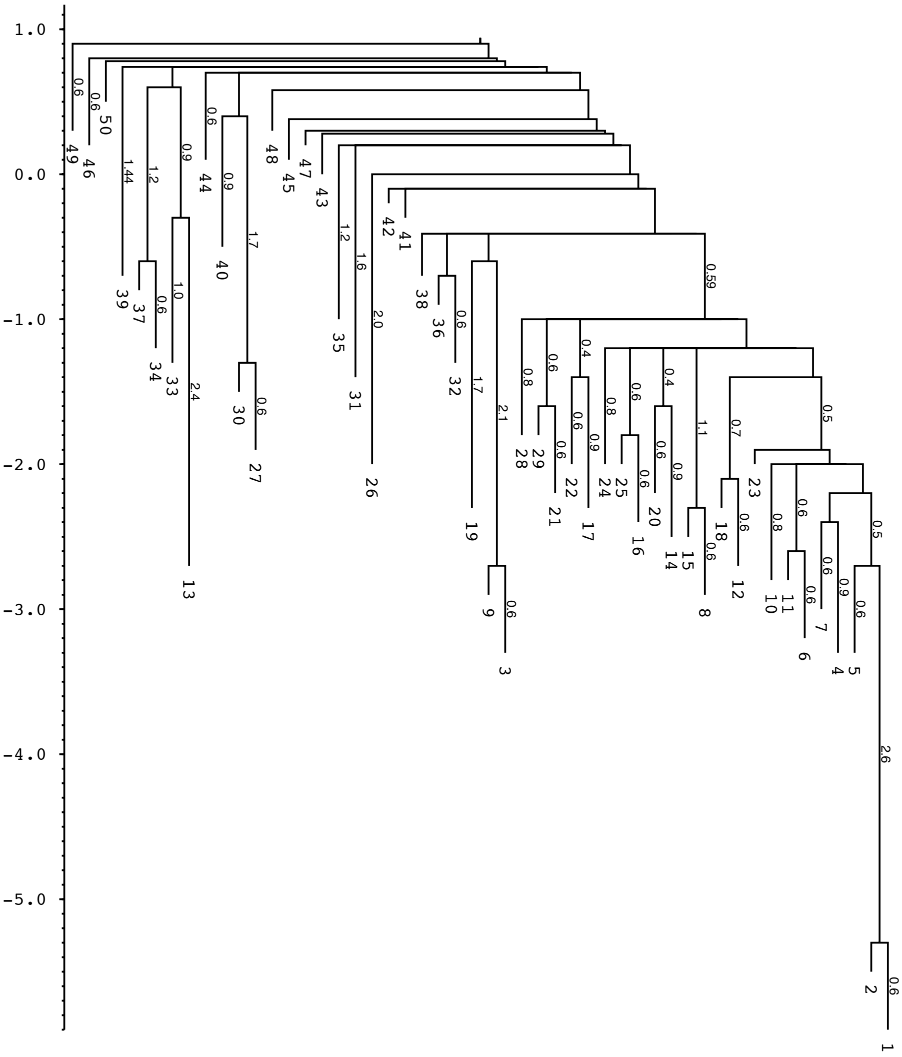

Figure 2.4: Barrier tree for the sequence subROSE (GUACCCAUCUUGCUCCU-UGGAGGAUUUGGCUAU)

2.3 RNA Metrics

In some applications we need to get estimations of likelihood of RNA structures. To do this we can use different metrics. The simplest example is the *structural distance metric.*

Definition 2.3.1 (Structural distance metric)**.** Let B_S, where S is a RNA structure, be a set of base pairs of S. Then the *structural distance* between two RNA structures S_1 and S_2 equals the symmetric difference size of B_{S_1} and B_{S_2}.

$$d_S(S_1, S_2) = |(B_{S_1} \cup B_{S_2}) \setminus (B_{S_1} \cap B_{S_2})|$$

An example of a more accurate metric is *mountain metric* (Hogeweg & Hesper, 1984; Moulton *et al.*, 2000) which allow to capture more secondary structure information.

Definition 2.3.2 (Mountain metric)**.** For each RNA structure S of the length n we define a vector f_S of the size n as follows: $f_S(i)$ equals the number of "(" brackets minus the number of ")" brackets found when looking through the bracket notation from the first position up to, and including, the i-th position so that $f_S = (f_S(1), f_S(2), \ldots, f_S(n))$. Furthermore let

$$w_S(k) = \begin{cases} \frac{1}{l-k} & \text{if } (k,l) \in B_S \\ \frac{-1}{k-l} & \text{if } (l,k) \in B_S \\ 0 & \text{otherwise} \end{cases}$$

and $f'_S(i) = \sum_{k=1}^{i} w_S(k)$ and $d_M(S_1, S_2) = ||f'_{S_1} - f'_{S_2}||_1 = \sum_{i=1}^{n} |f'_{S_1}(i) - f'_{S_2}(i)|$.

2.4 Abstract Shapes of RNA

Unfortunately the size of the state space of the energy landscape grows exponentially in the size of RNA sequence. Thus one of the important questions is to find the appropriate abstraction of the landscape. One of the approaches is discussed in (Giegerich *et al.*, 2004; Steffen *et al.*, 2006; Reeder & Giegerich, 2005).

According to Steffen *et al.* (2008) five different level of abstractions are defined. The difference between types of abstraction is illustrated with the following example structure: (((((((.((((.((......)).((.((.......)).)).)))).((((.......)))).))))).))..

- Type 1 (Most accurate): all loops and all unpaired regions are represented. All structural components contribute to shape representation, only the length of loops and unpaired regions is abstracted.

  [[_[_[]_[_[]_]_]_[]_]_]_

- Type 2: nesting pattern for all loop types and unpaired regions in external loop and multiloop.

  [[[[][_[]_]][]]_]

- Type 3: nesting pattern for all loop types but no unpaired regions. This shape representation completely abstracts from single-stranded regions.

  [[[[][[]]][]]]

- Type 4: helix nesting pattern and unpaired regions in external loop and multiloop. In this type helices are combined and thus we additionally abstract from nesting and adjacency of helices.

  [[[][[]]][]]

- Type 5 (Most abstract): helix nesting pattern and no unpaired regions.

  [[[][]][]]

Now we will formally define level of abstraction 5. We will call π a mapping from the tree-like domain of concrete structures to the tree-like domain of abstract structures. The representative structure $\hat{p}$ for shape class p is the element that has minimal free energy among all structures in the class (we will call such a structure a *shrep*). Due to Zucker energy model RNA structure consists of the following components (see Definition 2.1.2): single-stranded regions (SS), hairpin loops (HL), stacking regions (SR), bulges on the 5' or on the 3' side (BL and BR), internal loops (IL) and multiloops. Furthermore we could have a list of adjacent structures (AD) and empty list of adjacent structures (E). We also need to introduce the notion for shape domain. We will do it as follows: OP - open structure, CL - closed structure, FK ('fork') - branching. Now we can formally define π:

$$\begin{array}{lcr}
\pi(SS(l)) & = & OP \\
\pi(HL(a,l,b) & = & CL \\
\pi(SR(a,x,b)) & = & \pi(x) \\
\pi(BL(a,l,x,b)) & = & \pi(x) \\
\pi(BR(a,x,l,b)) & = & \pi(x) \\
\pi(IL(a,l,x,l',b)) & = & \pi(x) \\
\pi(ML(a,c,b) & = & FK(\pi(c)) \\
\pi(AD(SS(l),c)) & = & \pi(c) \\
\pi(AD(x,c)) & = & AD(\pi(x),\pi(c)) \text{ for } x \neq SS(l) \\
\pi(E) & = & E
\end{array}$$

This abstraction function retains hairpins and multiloops, but abstracts from stack lengths, bulges, internal loops and single-stranded regions (except in the case of the completely unpaired structure). In this manner one can formally define other levels of abstraction. For more information see Giegerich *et al.* (2004). In this work will consider shapes of RNA structures without lonely base pairs (i.e., pairs that are not stacked on another pair). But to be able to work with RNA structures with lonely base pairs we will use the following transformation: we delete all lonely-standing base pairs and then associate the shape of the structure we got with the original RNA structure.

Till now we have only defined tree based representations for abstract and concrete domains. For convenience we will introduce string based representations of both abstract and concrete domains.

We define a notation for shapes, using mapping ν_P as follows: $\ldots_k$ means k dots, $|l|$ is the length of string l and ε denotes the empty string.

$$
\begin{array}{lcr}
\nu_P(OP) & = & \_ \\
\nu_P(CL) & = & [] \\
\nu_P(FK(c)) & = & [\nu_P(c)] \\
\nu_P(AD(x,c)) & = & \nu_P(x)\nu_P(c) \\
\nu_P(E) & = & \varepsilon
\end{array}
$$

The notation for the concrete domain is similar to dot-bracket representation (see Definition 2.1.3), here defined as ν_S:

$$
\begin{array}{lcr}
\nu_S(SS(l)) & = & \ldots_l \\
\nu_S(HL(a,l,b)) & = & (\ldots_l) \\
\nu_S(SR(a,x,b)) & = & (\nu_S(x)) \\
\nu_S(BL(a,l,x,b)) & = & (\ldots_{|l|}\,\nu_S(x)) \\
\nu_S(BR(a,l,x,b)) & = & (\nu_S(x)\ldots_{|l|}) \\
\nu_S(IL(a,l,x,l',b)) & = & (\ldots_{|l|})\nu_S(x)\ldots_{|l|}) \\
\nu_S(ML(a,x,b)) & = & (\nu_S(x)) \\
\nu_S(AD(x,c)) & = & \nu_S(x)\nu_S(c) \\
\nu_S(E) & = & \varepsilon
\end{array}
$$

Chapter 3

Exact methods

3.1 Flooding Algorithm for Barriers

In (Kubota & Hagiya, 2005) a general approach for finding barriers between structures is proposed. This algorithm is an implementation of the idea of flooding algorithm. In the algorithm the energy landscape is represented as a graph $G = (V, E)$, where V is a conformation space and the set of edges E is defined using move set. Pseudocode for flooding algorithm for barriers is presented in Algorithm 1.

Algorithm 1 Flooding Algorithm for Barriers

```
$S_s$                              ▷ initial structure
$S_t$                              ▷ target structure
$\mathcal{B}$                      ▷ set of reachable, low energy vertices
$\mathcal{N}$                      ▷ set of vertices neighboring a vertex in $\mathcal{B}$
$\mathcal{M}$                      ▷ set of vertices which were added on the current interation
$\mathcal{N} \leftarrow \emptyset$
$\mathcal{B} \leftarrow \{S_s\}$
$\mathcal{M} \leftarrow \{S_s\}$
while $S_t \notin \mathcal{M}$ do
    $\mathcal{N} \leftarrow \mathcal{N} \cup \{\text{neighbours of } v \mid v \in \mathcal{M}\} \setminus (\mathcal{B} \cup \mathcal{N})$
    $\mathcal{M} \leftarrow \{\hat{x} \in \mathcal{N} \mid E(\hat{x}) = \min\{E(x) \mid x \in \mathcal{N}\}\}$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{M}$
end while
```

The structure with the maximum energy in $\mathcal{B}$ is a true energy barrier between initial and target structures. Unfortunately, in the worst case we need to enumerate the whole structure space, i.e. we need exponential time in the length of the input RNA sequence.

3.2 Dynamic Programming Approach for Direct Paths

In order to decrease the number of structures under consideration we will abstract the landscape as follows: we will group structures depending on the structural distance from initial and target structures. We will use structural distance metric. We should mention that only direct paths from initial to target structure are considered in the following approach. In Algorithm 2 the pseudocode of DP approach is presented.

Algorithm 2 Dynamic programming approach

```
$S_s$                                      ▷ initial structure
$S_t$                                      ▷ target structure
$C_i$                                      ▷ $C_i = \{s|d_S(s, S_s) = i \wedge d_S(s, S_t) = dist - i\}$
        ▷ i.e., the set of structures which are in the distance of i to the initial state
        ▷ and $dist - i$ to the target state
$B_i$            ▷ Barriers for the path which ends in class $C_i$. $B_i(struct)$ represents
        ▷ a barrier for the path ending in structure $struct$ in class $C_i$
$path$                                     ▷ path between initial and target structures
$Barrier \leftarrow Infinity$
$dist \leftarrow d_S(S_s, S_t)$
Initialization of $C_1$ and $B_1$
for $i = 2 \ldots dist$ do
    for all $curr \in C_i$ do
        for all $prev \in C_{i-1}$ do
            $Barrier \leftarrow max(B_{i-1}(prev), Energy(curr))$
            $B_i(curr) \leftarrow min(Barrier, B_i(curr))$
        end for
    end for
end for
Output  $B_{dist}$(target state)           ▷ Barrier between initial and target structures
$path \leftarrow BackTrack(B_{dist}$(target state))   ▷ We get the path between $S_s$ and $S_t$
        ▷ using backtracking
```

Chapter 4

Heuristics

The algorithms which were considered in Chapter 3 give the exact results but are not applicable to long sequences. To overcome this obstacle several heuristics have been developed. This chapter gives overview of already existing heuristic approaches and present some new algorithms.

4.1 Morgan Higgs Heuristic

One of the most important and common heuristics to find barriers in the landscape is Morgan-Higgs heuristic (Morgan & Higgs, 1998). Now we will briefly describe the underlying algorithm. Algorithm 3 presents pseudocode for Morgan-Higgs heuristic.

The Morgan-Higgs heuristic aims at determining the barrier between two conformations A and B. It only considers direct walks between A and B. To introduce Morgan-Higgs heuristic we need one more definition.

Definition 4.1.1 (Conflicting base pairs)**.** Let S be an RNA sequence and P_1 and P_2 be two structures of S. Then $p \in B_{P_1}$ is in conflict with $q \in B_{P_2} \setminus B_{P_1}$ if in order to add q to P_1 one should first delete p from P_1.

Algorithm 3 Morgan Higgs Heuristic

```
  $A_{add} \leftarrow B \backslash A$                    ▷ the base pairs to add to get from $A$ to $B$
  $A_{remove} \leftarrow A \backslash B$                 ▷ the base pairs to remove to get from $A$ to $B$
  Sort $A_{add}$ by ascending number of conflicting base pairs with $A_{remove}$
  for all basepair $p \in A_{add}$ do
      Remove from the structure the base pairs from $A_{remove}$ which are in conflict
      with $p$
      Add all elements in $A_{add}$ without conflicts to the structure
      Record the the structure with the maximum energy over all structures we got
      after deleting some conflict base pairs and adding new base pairs in
      the previous two steps
  end for
```

We also need to take the following remarks into consideration:

1. The Morgan-Higgs heuristic returns the energy barrier of the lowest traversed path. There is no guarantee that the choice of routes includes the lowest direct route.

2. When there are several base pairs with an equal number of conflicts, paths for each possible ordering may be calculated in order to get better results.

4.2 Breadth First Search

In Section 3.2 we considered a method to exactly calculate the barrier when we take only direct paths into consideration. The disadvantage of this method is in its complexity. To overcome this obstacle we consider the following method. This approach works as follows:

1. We start in the initial structure. We generate all neighbored structures of the initial structure which are in the next distance class (in this case class $(1, dist - 1)$). Thus we will get partial paths of the length 2.

2. We calculate barriers for each of these paths. We save $MaxKeep$ best structures.

3. We generate all neighbor structures for the set of structures we got in the previous step. We proceed in the same manner as in steps 1 and 2 until we reach the final structure.

Algorithm 4 presents pseudocode for the breadth first search.This approach was first introduced in Flamm *et al.* (2001).

Algorithm 4 Breadth first search

```
$S_s$                                                    ▷ initial structure
$S_t$                                                    ▷ target structure
$S$                              ▷ the set of structures under consideration
                                ▷ each element also contains information about
            ▷ previous state on the partial path and the current values of barrier
$next$                                      ▷ the set of neighbor structures
$MaxKeep$                          ▷ number of structures to keep on each step
$path$                            ▷ path between initial and target structures
$S \leftarrow \{S_s\}$
$dist \leftarrow$ structural distance between initial and target states
for $i = 1 \dots dist - 1$ do                          ▷ for all distance classes
    $next \leftarrow Neighbors(S, i)$     ▷ all neighbors in the next distance class
    $S \leftarrow KeepBest(next, MaxKeep)$
end for
Output $min(S)$                   ▷ Barrier between initial and target structures
$path \leftarrow BackTrack(argmin(S))$
```

Remarks:

1. $C_i = \{s | d_S(s, S_s) = i \wedge d_S(s, S_t) = dist - i\}$ – the set of structures which are in the distance of i to the initial state and $dist - i$ to the target state.

2. $Neighbors(S, i)$ – is a function which returns a set $\{s' | s' \in C_i \wedge \exists s \in S : d_S(s, s') = 1\}$, i.e. the structures which lie in the next distance class and are neighbored to some structure in S.

3. $KeepBest(next, MaxKeep)$ – is a function which returns $MaxKeep$ structures with minimal energy from the set $next$.

4. $BackTrack(argmin(S))$ – is a function which prints out the path with minimal maximal energy between S_s and S_t using backtracking.

In order to improve performance of BFS we consider a modification of BFS method. We will order the structures in the distance classes in specific way using mountain metric and partial barrier values. To do it we need to modify the $KeepBest(next, MaxKeep)$ function. Let $struct \in C_i$. We present the following weighting function:

$$score(struct) = w_B \cdot score_{barrier}(struct) + w_M \cdot score_{mountain}(struct),$$

where

w_B and w_M – weights of mountain metric and partial barrier respectively,

$w_B + w_M = 1, w_B \geq 0, w_M \geq 0,$

$$score_{barrier}(struct) = \frac{barrier(struct) - min_{barrier}(struct)}{max_{barrier}(struct) - min_{barrier}(struct)},$$

$$score_{mountain}(struct) = \frac{d_M(struct, S_t) - min_{mount}}{max_{mount}(struct) - min_{mount}(struct)},$$

$barrier(struct)$ – partial barrier till $struct$,

$min_{barrier}(struct)$ – minimal partial barrier in the distance class C_i,

$max_{barrier}(struct)$ – maximal partial barrier in the distance class C_i,

$min_{mount}(struct)$ – minimal mountain metric value in the distance class C_i,

$max_{mount}(struct)$ – maximal mountain metric value in the distance class C_i.

We will sort the structures in the set $next$ using this scoring function and after that take $MaxKeep$ best.

4.3 Shape Network

The main disadvantage of a distance abstraction is the large similarity of neighbored distance classes. Furthermore when using a distance classes approach we cover only a small part of the state space. We will try to overcome this obstacles by using shapes abstraction. We have already defined shape abstraction in chapter 2.4. The Shape Network algorithm works as follows:

1. Using RNAshapes (Steffen *et al.*, 2008) we can compute a list of all possible shapes; each shape except initial and target RNA structures is represented by a shrep (see Section 2.4); shape to which initial RNA structure belongs to (we call such shape *initial shape*) is represented by an initial RNA structure; the same for target RNA structure (we call such shape *target shape*).

2. Using BFS (or MH) we can compute barriers between all pairs of shapes. We save this data in the matrix (we call this barrier matrix). Thus we get a graph where a vertex represents a shape and the weight of an edge is equal to the barrier height between vertices of the edge.

3. Using modification of Floyd-Warshall algorithm (Floyd, 1962) we calculate barrier between initial and target shapes.

The pseudocode of the modified Floyd-Warshall algorithm is presented in Algorithm 5.

The algorithm can also be modified in the following way:

1. initial and target shapes are represented by their shreps.

2. same as before.

3. same as before.

4. using BFS (or MH) we calculate barrier between initial structure and the shrep of the initial shape; the same for target structure.

5. we get final barrier using matrix calculated in 3 and barriers from 4.

Using this modified algorithm we can effectively get approximations of barriers for all pairs of structures and do not need to recalculate the barrier matrix. Thus this algorithm becomes applicable to problems where we need to calculate multiple times barriers between different pairs of structures in the same landscape (the same RNA sequence). As an example of such a problem we can mention the problem of computating a barrier tree (Richter, 2007).

Algorithm 5 Modified Floyd algorithm for calculating barriers

```
$i, j, k$
$dist$                                  ▷ matrix of barriers along direct paths between shreps
$back$                                  ▷ data for backtracking
$N$                                     ▷ number of shapes in shape network
$init\_shape$                           ▷ initial shape
$target\_shape$                         ▷ target shape
$path$                                  ▷ path between initial and target structures
for $k = 1 \dots N$ do
    for $i = 1 \dots N$ do
        for $j = 1 \dots N$ do
            $curr\_barr \leftarrow dist(i, j)$
            $new\_barr \leftarrow max(dist(i, k), dist(k, j))$
            if $new\_barr < curr\_barr$ then
                $dist(i, j) \leftarrow new\_barr$
                $back(i, j) \leftarrow k$
            end if
        end for
    end for
end for
Output $dist(init\_shape, target\_shape)$     ▷ Barrier between initial and target
                                              ▷ structures
$path \leftarrow BackTrack(init\_shape, target\_shape)$
```

4.4 Shape Triples Approach

In this section a method to decrease time complexity of Shape Network Method is presented. To do this we consider only the paths of the form:

initial structure - *shrep* - target structure.

Our hypothesis is that to get good results we do not need to consider the paths with the complex structure. In this approach we will consider the paths which consists of two parts and each of them is a direct path as well as direct path between initial and target structure. As before the shape to which the initial RNA structure belongs to is represented by an initial RNA structure (target shape is represented by the target RNA structure). Algorithm 6 presents pseudocode of the Shape Triples Approach.

Remarks:

1. $CalcPath(structA, structB)$ – calculates barrier height between $structA$ and $structB$. We can use either BFS or MH.

One can also consider a modification of Shape Triples approach in which we consider not a single shrep for each shape but a set of structures with the minimal energy from the shape. We will call this method *Shape Triples with Sets*.

Algorithm 6 Shape Tripples Approach

```
$i$
$shreps$                                                   ▷ array of shreps
$best\_i$                                                  ▷ best shrep
$best\_barrier$                                            ▷ best barrier
$current\_barrier$                                         ▷ current barrier
$N$                                      ▷ number of shapes in shape network
$init\_shape$                                              ▷ initial shape
$target\_shape$                                            ▷ target shape
$path$                              ▷ path between initial and target structures
$best\_i \leftarrow -1$      ▷ in the case when we do not have any intermediate shapes
$best\_barrier \leftarrow CalcPath(init\_shape, target\_shape)$    ▷ as $CalcPath$ we can use
                                                           ▷  either BFS or MH
for $i = 1 \ldots N$ do
    $dist\_init \leftarrow CalcPath(init\_shrep, i)$        ▷ Calculate barrier between initial
                                                    ▷ structure and $i$-th shrep
    $dist\_target \leftarrow CalcPath(i, target\_shrep)$    ▷ Calculate barrier between $i$-th
                                                    ▷ shrep and target structure
    $current\_barrier \leftarrow max(dist\_init, dist\_target)$
    if $current\_barrier < best\_barrier$ then
        $best\_barrier \leftarrow current\_barrier$
        $best\_i \leftarrow i$
    end if
end for
Output $best\_barrier$                      ▷ Barrier between initial and target
                                                    ▷ structures
```

4.5 Direct Shape Paths

In the previous approaches in which we used shape abstraction we had to run through the whole list of shapes. In the Direct Shape Paths approach we want to consider only the shapes which are relevant to the path between given initial and target RNA structures. We will proceed as follows:

1. We calculate abstract shapes of the initial and target structure. We call the shape which includes initial structure *initial shape*. We call the shape which includes target structure *target shape*.
2. We find out the path in the abstract space between initial and target shapes. Neighborhood relation is defined as insertion or deletion of one bracket pair. We associate with each shape class except initial and target shapes its shrep. The energy of initial structure is associated with the initial shape. The energy of target structure is associated with the target shape. Finally we calculate the abstract path using modification of BFS. As *element of the abstract path* we understand a set of shape classes which have the same distances to initial and target shapes.
3. The path between structures will be small even for long concrete sequences.
4. We start considering initial shape. We generate shreps in the next shape class on the path. We could have a set of shreps because as mentioned above each element of the abstract path is a set of shapes which have the same distances to initial and target shapes.
5. We calculate the barrier between initial shape and each of concrete structures. We can do it using either BFS or MH.
6. Now we can calculate partial paths and partial barriers for these concrete structures.
7. We iterate through steps 5-6 until we reach the target shape.

Algorithm 7 presents pseudocode for the Direct Shape Paths approach.
Remarks:

1. $Shape(struct)$ – returns the shape of the structure $struct$.
2. $CalcPath(init\_shape, target\_shape)$ – returns a path between $init\_shape$ and $target\_shape$ in the space of shapes.
3. $GenRepr(abstract\_class)$ – returns a set of shreps of the shape classes $abstract\_class$.

4. $CalcPartialBarriers(prev\_class, curr\_class)$ – calculates partial barriers for the paths ending in $curr\_class$ using information from $prev\_class$ and saves this information in $curr\_class$.

5. $BackTrack(argmin(curr\_class))$ – returns the concrete path using backtracking.

Algorithm 7 Direct Shape Paths Approach

```
$S_s$                                                          ▷ initial structure
$S_t$                                                          ▷ target structure
$init\_shape$                                                  ▷ initial shape
$target\_shape$                                                ▷ target shape
$curr\_class$                                                  ▷ current class on the path
$prev\_class$                                                  ▷ previous class on the path
$abstract\_path$                               ▷ abstract path between initial and final shapes
$concrete\_path$                               ▷ path between initial and target structures
$init\_shape \leftarrow Shape(S_s)$
$target\_shape \leftarrow Shape(S_t)$
$abstract\_path \leftarrow CalcPath(init\_shape, target\_shape)$
$prev\_class \leftarrow \{S_s\}$
for all $abstract\_class \in abstract\_path[2, \ldots]$ do▷ all classes except the initial one
    $curr\_class \leftarrow GenRepr(abstract\_class)$                 ▷ we generate shreps
    $CalcPartialBarriers(prev\_class, curr\_class)$  ▷ calculate partial barriers and
                                                       ▷ save results in $curr\_class$
    $prev\_class \leftarrow curr\_class$
end for
Output $min(curr\_class)$                     ▷ Barrier between initial and target structures
$path \leftarrow BackTrack(argmin(curr\_class))$
```

Chapter 5

Experimental Results

In the Chapters 3 and 4 several methods for finding barriers were described. In this Chapter we will evaluate and compare the described methods.

5.1 Methodology of Experiments

The following three RNA sequences have been considered in the experimental part:

1. *subROSE* – `GUACCCAUCUUGCUCCUUGGAGGAUUUGGCUAU`

   This is a subsequence of ROSE Element (Chowdhury *et al.*, 2006).

2. tRNA of *Caenorhabditis brenneri* – Caenorhabditis_brenneri_chrUn.trna825-AlaAGC[1] (187465963-187465891)

   `GGGGGTATAGCTCAGTGGTAGAGCGCTCCCTTAGCATGGGAGAGGGCTGGGGTTCAATTCC-CCCATACCTCCA`

3. tRNA of *Chlamydia trachomatis* – Chlamydia_trachomatis_A_HAR-13_chr.trna21-AlaGGC[2] (728227-728155)

   `GGGGTATTAGCTCAGTTGGTAGAGCGCAACAATGGCATTGTTGAGGTCAGCGGTTCGATCCCG-CTATGCTCCA`

For each sequence `RNAsubopt` program from Vienna RNA Package[3] version 1.8.2 (Flamm *et al.*, 2002; Wolfinger *et al.*, 2004; Wuchty *et al.*, 1999) was executed. The program was run with the following parameters:

- *subROSE* – `RNAsubopt -e 20 -d2 -s` (`-d2` means that dangling energies will be added for the bases adjacent to a helix on both sides and `-e 20` means that suboptimal structures withing 20 kcal/mol of the minimum free energy (mfe) structure will be calculated, `-s` means that the structures will be sorted in the increasing order according to their energy).

[1] The sequence was taken from http://gtrnadb.ucsc.edu/Cbren/

[2] The sequence was taken from http://gtrnadb.ucsc.edu/GtRNAdb/Chla_trac_A_HAR-13/

[3] Vienna RNA Package can be downloaded for free from http://www.tbi.univie.ac.at/RNA/

- *Caenorhabditis brenneri* – `RNAsubopt -e 22.2 -d2 -s`
- *Chlamydia trachomatis* – `RNAsubopt -e 25 -d2 -s`

After that the results were forwarded to `barriers` program[4] version 1.5.2 with the following parameters: `barriers -G RNA -M noShift` (`-G RNA` means that we consider RNA structures, `-M noShift` means that we use single move set (see Section 2.2)).

The output contained a list of pairs of local minima and exact barriers between them. This list of pairs of local minima was used as an input of heuristics which are experimentally considered in this chapter.

To produce plots we used R[5] (R. D. C. Team, 2004). The kcal/mol is used as a measure unit in plots which present barriers' estimations between structures or the difference between approximated and exact barriers.

The considered heuristics were implemented in C++ using the Energy Landscape Library[6] (Mann *et al.*, 2007).

5.2 Distance abstraction

5.2.1 subROSE

In this section we will consider the sequence *subROSE* of the length 33.

On Figure 5.1 we can see the difference between approximated and exact barriers for subROSE sequence. We consider the following algorithms: dynamic programming approach (Algorithm 2, in Figure 5.1 referenced as DP), breadth first search approach (Algorithm 4; $MaxKeep = 5$, i.e. we keep 5 structures at each step; in Figure 5.1 this algorithm is referenced as BFS) and Morgan-Higgs heuristic (Algorithm 3, in Figure 5.1 referenced as MH). From this figure we can conclude that we can get the best results using dynamic programming algorithm (which is unfortunately infeasible in practice because of the exponential blow up of number of structures in structural distance classes). Morgan-Higgs heuristic gives us the worst results.

Figure 5.2 represents the distribution of differences between approximated and exact barriers according to structural distance between initial and target structures. We can see that we get worse approximations when we consider the structures with large structural distance for all considered methods. We would like to emphasize that the results of MH heuristic crucially depend on the structural distance. In the case of large structural distance we get very over-approximated results.

From Figure 5.1 we can see that deviation of DP algorithm is very small (in particularly in comparison with MH heuristic) but still non-zero. Thus a question

[4] `barriers` program can be downloaded for free from http://www.tbi.univie.ac.at/ ivo/RNA/Barriers/

[5] R can be downloaded for free from http://www.tbi.univie.ac.at/RNA/

[6] ELL can be downloaded for free from http://www.bioinf.uni- freiburg.de/SW/ELL/

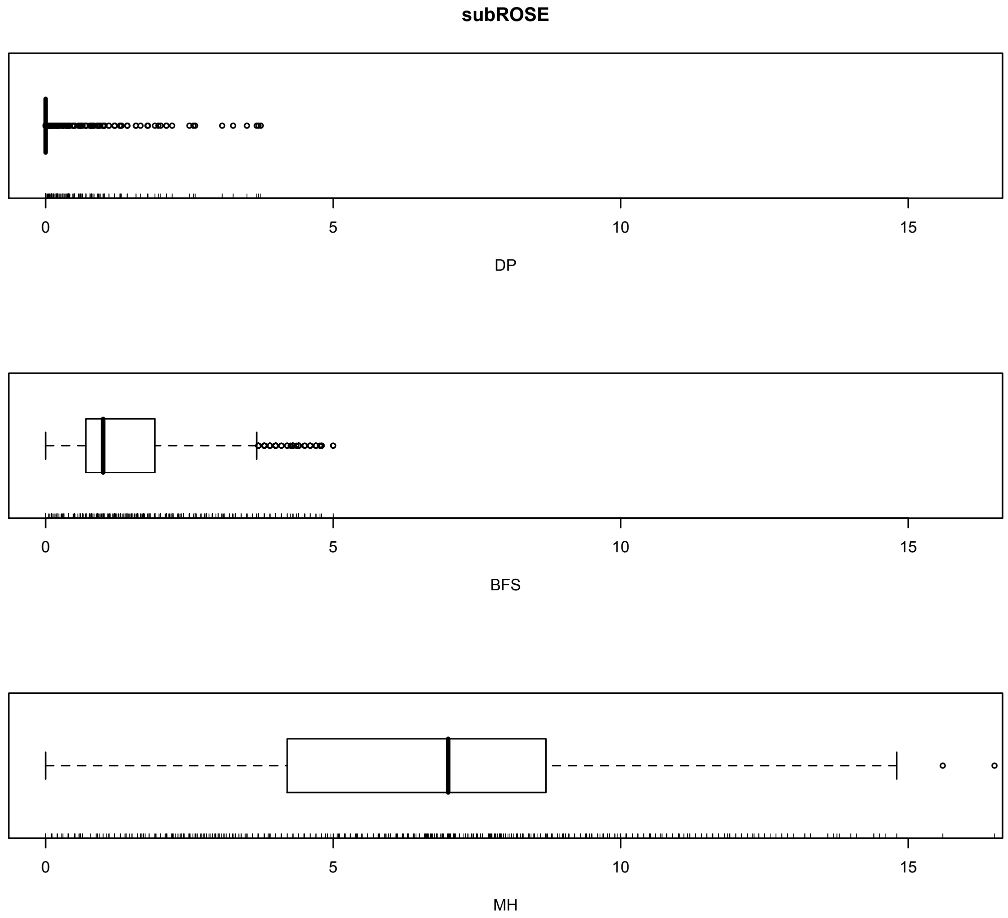

Figure 5.1: Difference between approximated and exact barriers for subROSE sequence. The results are represented using box-and-whisker plot. The following data is visualized: smallest non-outlier observation (tick in the left part), lower quartile (left border of the box), median (line dividing the box), upper quartile (right border of the box), largest non-outlier observation (tick in the right part), outliers (dots)

appears weather it is sufficient to consider only direct paths between initial and target structures. To investigate this question we conducted two more tests. We calculated optimal paths between structures and then researched the structure of optimal paths.

Figure 5.3 shows us the structure of optimal paths for pairs of initial and target structures with structural distance equal 10. We can see that there are a lot of optimal paths which go through classes on the direct path. But nevertheless we have a lot of paths which have classes far from direct path as their part. The situation is illustrated formally in Figure 5.5. It shows us the distribution of optimal paths according to paths' length between structure with structural distance equals 10. We can see that 23% of optimal paths are direct. Furthermore, optimal paths with the length less or equal 16 constitute 68% of all optimal paths with structural distance 10.

Figure 5.4 describes the structure of optimal paths for pairs of initial and target structures with structural distance equal 16. Using this figure we can get more

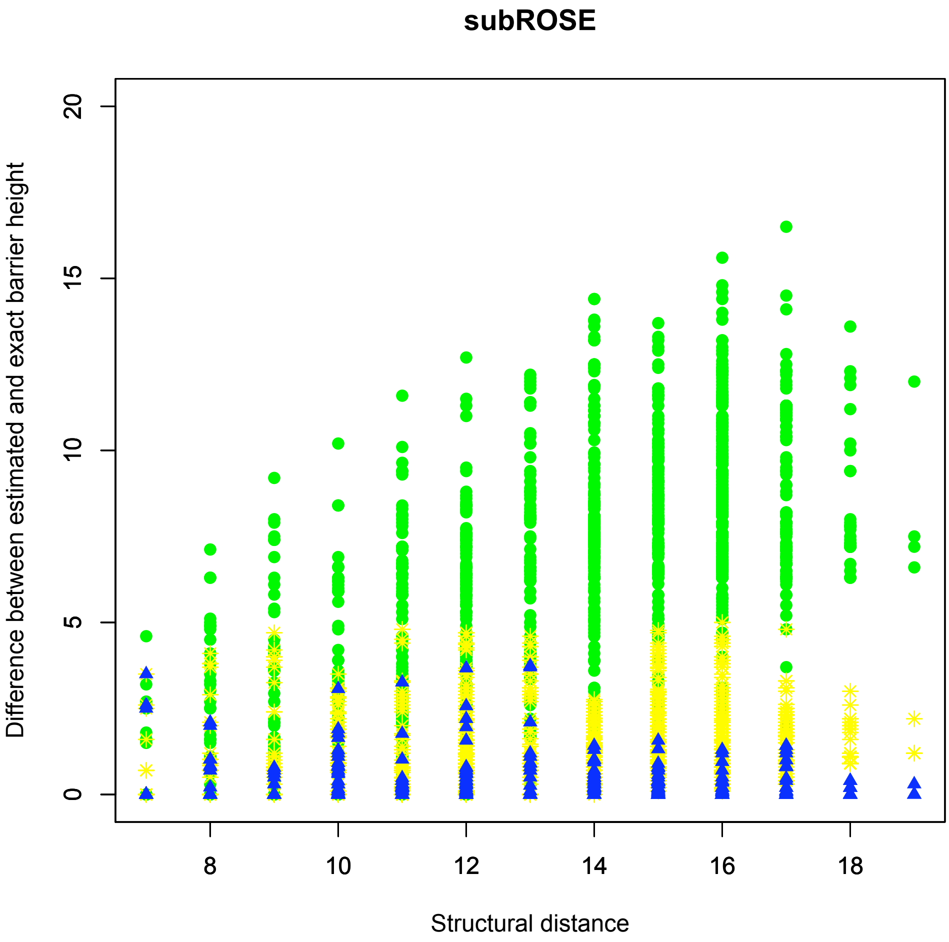

Figure 5.2: subROSE - Distribution of differences between approximated and exact barriers over structural distances. Blue triangles correspond to DP, yellow stars correspond to BFS and finally green circles represent results of MH.

insight in the structure of optimal paths. We can conclude that when we have larger structural distance that it is more probable to have a path far away from the direct one. Figure 5.6 gives us the numerical presentation of optimal path distribution with structural distance 16. In this case 23% of paths are direct and 46% of optimal paths has the length less or equal to 20.

5.2.2 Chlamydia trachomatis

It would be interesting to consider the behavior of the algorithms on larger sequences. In this chapter we will consider the sequence *Chlamydia trachomatis* of the length 73.

Figure 5.7 represents the difference between approximated and exact barriers for sequence *Chlamydia trachomatis*. The following algorithms are considered: dynamic programming approach, breadth first search approach and Morgan-Higgs heuristic. We can see that the approximation we got is worse then for *subROSE* sequence. Thus we can conclude that we get worse approximation for the longer sequences.

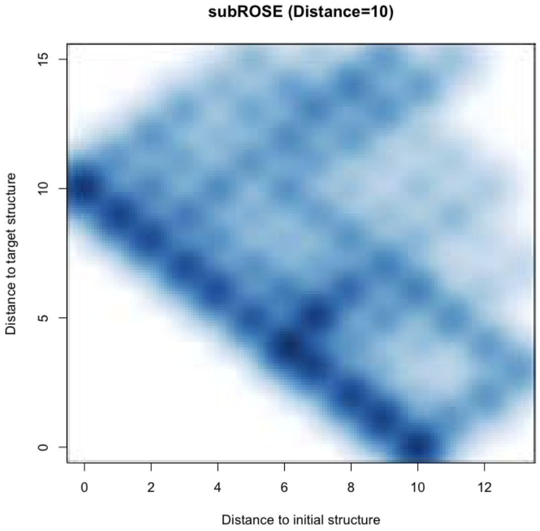

Figure 5.3: subROSE - Structure of optimal paths (Distance = 10)

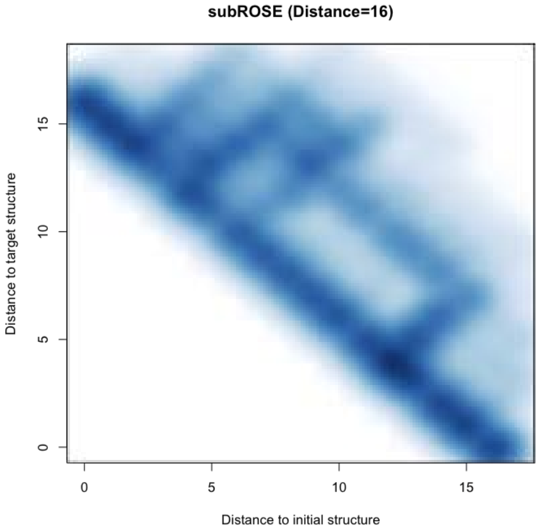

Figure 5.4: subROSE - Structure of optimal paths (Distance = 16)

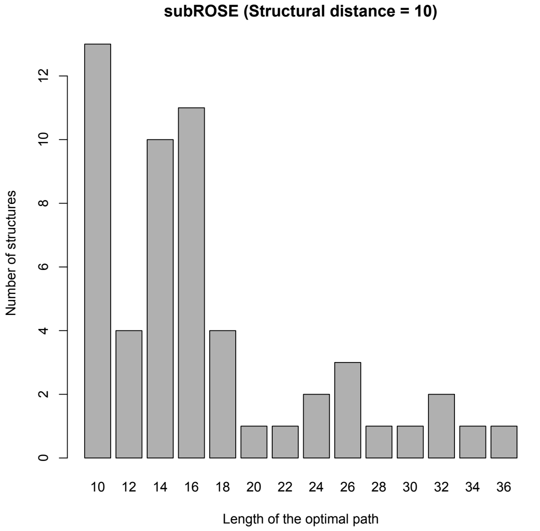

Figure 5.5: subROSE - Distribution of optimal paths according to paths' length
(Structural distance = 10)

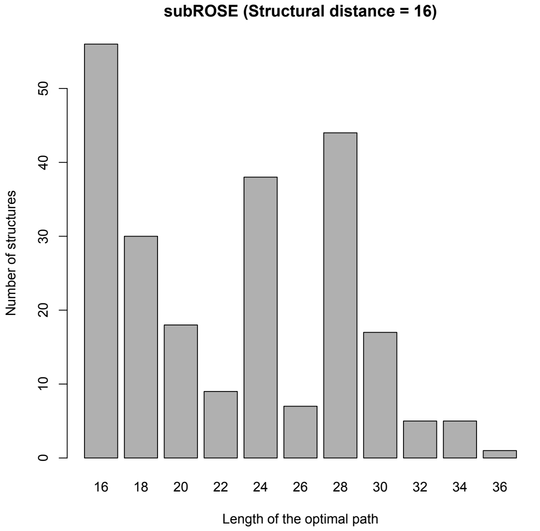

Figure 5.6: subROSE - Distribution of optimal paths according to paths' length
(Structural distance = 16)

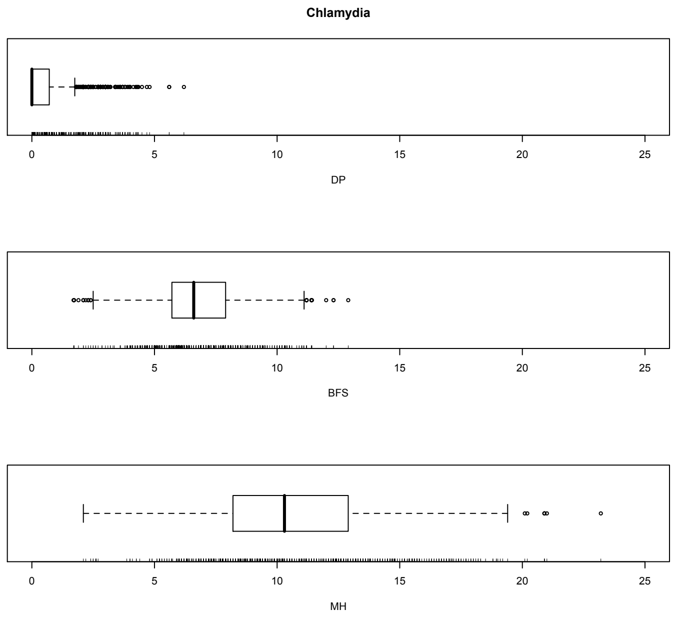

Figure 5.7: Difference between approximated and exact barriers for Chlamydia trachomatis

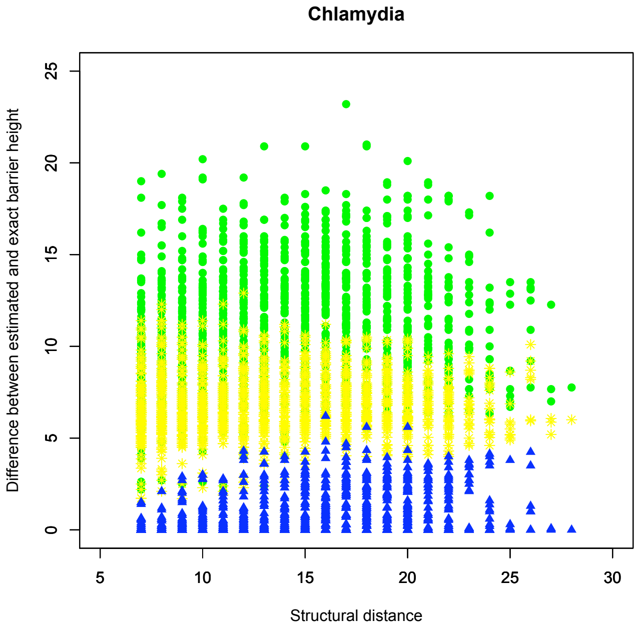

Figure 5.8: Chlamydia trachomatis - Distribution of differences between approximated and exact barriers over structural distances

Figure 5.8 represents the distribution of differences between approximated and exact barriers according to structural distance between initial and target structures. We can see that the results of all algorithms becomes worse when we consider the structures which are far away from each other.

From Figure 5.7 we can see that deviation of DP algorithm is larger then for *subROSE* sequence. Thus we can conclude that for very long sequences even DP will not be sufficient.

Finally, Figure 5.9 shows us that the optimal paths can have a much more com-

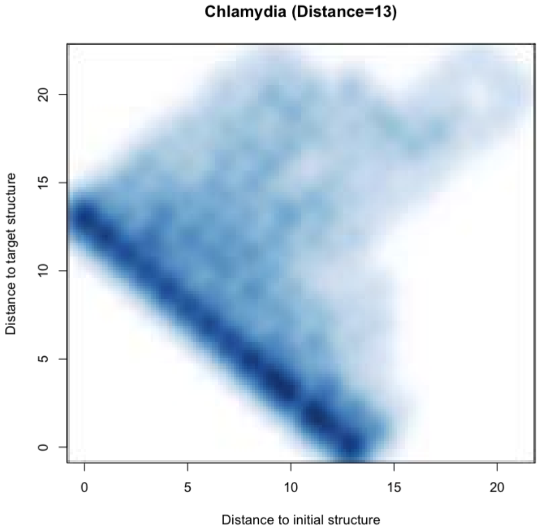

Figure 5.9: Chlamydia trachomatis – Structure of optimal paths (Distance = 13)

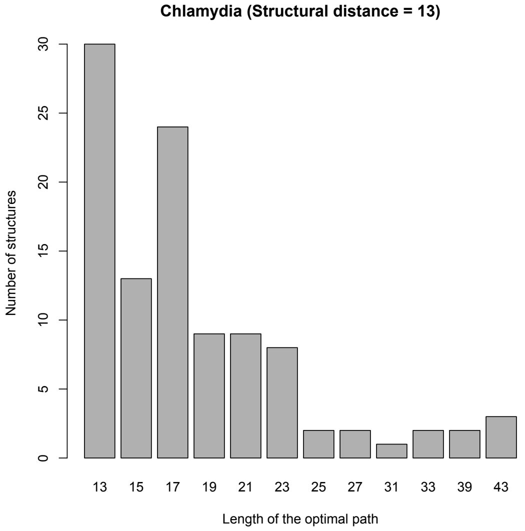

Figure 5.10: Chlamydia trachomatis – Distribution of optimal paths according to paths' length (Structural distance = 13)

plex structure in comparison with Figure 5.3. From Figure 5.10 we can find out that only 28% of optimal paths are direct and 64% of paths have the length less or equal to 17. It is the reason why it is not enough considering only direct paths for the sequence *Chlamydia trachomatis*.

5.2.3 Caenorhabditis brenneri

One more sequence which we consider is the sequence *Caenorhabditis brenneri* of the length 73.

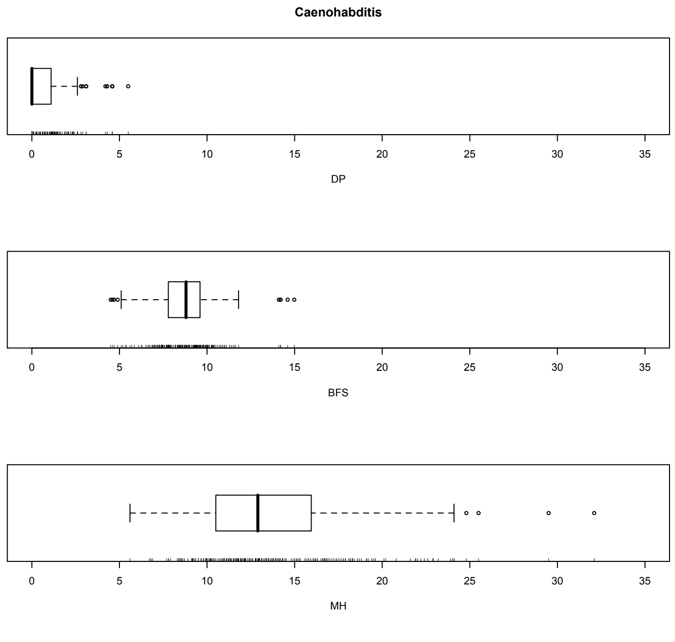

Figure 5.11: Difference between approximated and exact barriers for Caenorhabditis brenneri sequence

Figure 5.11 presents the difference between approximated and exact barriers for *Caenorhabditis brenneri* sequence. The following algorithms are considered: dynamic programming approach, breadth first search approach and Morgan-Higgs heuristic.The results of DP and BFS are quite similar to those for *Chlamydia trachomatis.* But MH approximates barriers of *Caenorhabditis brenneri* worse then barriers of *Chlamydia trachomatis.*

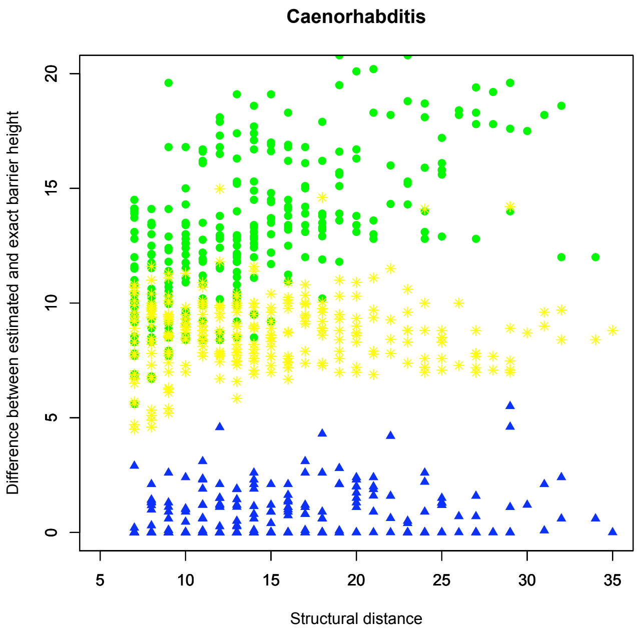

Figure 5.12: Caenorhabditis brenneri - Distribution of differences between approximated and exact barriers over structural distances

Figure 5.12 represents the distribution of differences between approximated and exact barriers according to structural distance between initial and target structures.

Finally, Figure 5.13 describes the structure of optimal paths with structural distance 13 and Figure 5.14 shows distribution of optimal paths with structural distance 13 over the length of paths. In the case of *Caenorhabditis brenneri* 55% of optimal paths are direct and 74% have the length less or equal to 19.

5.3 Mountain Metric

In Section 2.3 we introduced Mountain Metric. Furthermore, in Section 4.2 we described a modification of BFS which uses Mountain Metric to reorder structures in the class. Now we will evaluate this approach. In Figure 5.15 the results of applying BFS with mountain metric and the size of distance classes equal to 5 are presented. The following combinations of weights are considered:

1. Barrier weight = 1, Mountain weight = 0

2. Barrier weight = 3/4, Mountain weight = 1/4

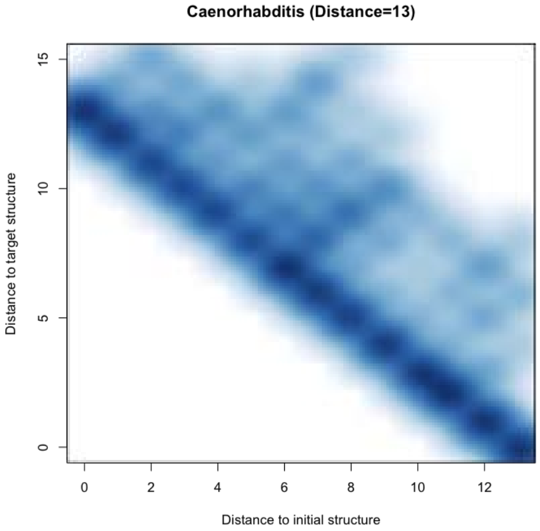

Figure 5.13: Caenorhabditis brenneri – Structure of optimal paths (Distance = 13)

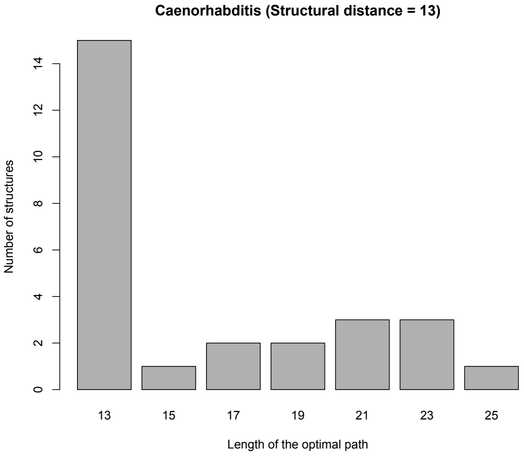

Figure 5.14: Caenorhabditis brenneri - Distribution of optimal paths according to paths' length (Structural distance = 13)

3. Barrier weight = 1/2, Mountain weight = 1/2

4. Barrier weight = 1/4, Mountain weight = 3/4

5. Barrier weight = 0, Mountain weight = 1

From this plot we can conclude that the more weight the mountain metric has the worse results we get. Thus mountain metric is not appropriate for estimating structure similarity in the case of barrier heights.

5.4 Shape abstraction

From Section 5.2 we can conclude that a lot of optimal paths are not direct. Thus it worth considering another distribution of structures in classes. In this section we will evaluate several algorithms based on abstract shapes of RNA, which was introduced in Section 2.4. As inputs we will use the data set, which we got using steps in Section 5.1.

5.4.1 Structure of optimal paths

In Figures 5.16 and 5.17 the distribution of optimal paths into shape classes for the paths with shape distance 4 and abstraction level 2 and 3 respectively is presented. We can see that the most of optimal paths fall onto direct shape paths. Furthermore when we consider coarser level of abstraction (in our case level 3 in comparison to level 2) we note the more structures are on the direct shapes paths. Thus we can

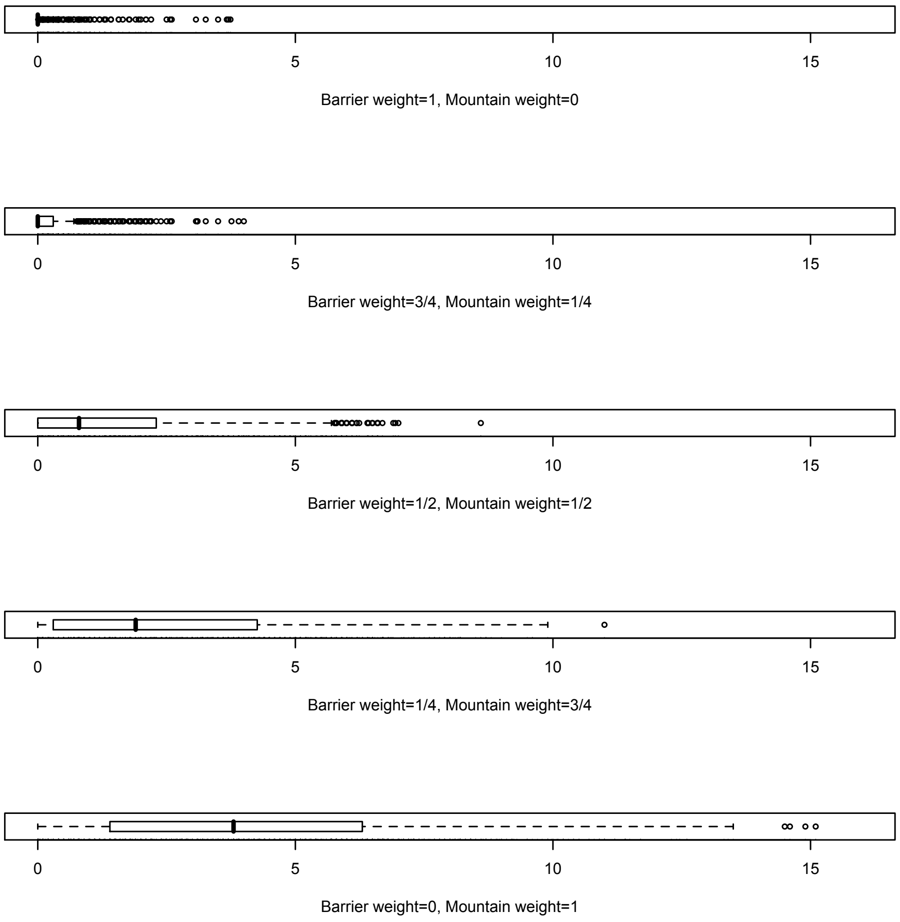

Figure 5.15: subROSE -Mountain metric (BFS, MaxKeep=5)

conclude that a shape abstraction gives us a good approximation of path space between initial and target structures.

5.4.2 Shapes Network

In Figure 5.18 the results of applying Shapes Network approach (see Section 4.3) in combination with MH heuristics onto *subROSE* sequence are presented. Figure 5.20 presents the results of applying Shapes Network approach in combination with BFS ($MaxKeep = 5$) onto *subROSE* sequence. One can point out that we get better results using Shapes Network approach then in the case of both BFS and MH. We want to notice the following fact from Figure 5.18: we get better results using shape abstraction level 4 in comparison to the level 5, but we get worse results using abstraction level 3 in comparison with abstraction level 4. This fact follows from the non-monotonicity of shape abstraction.

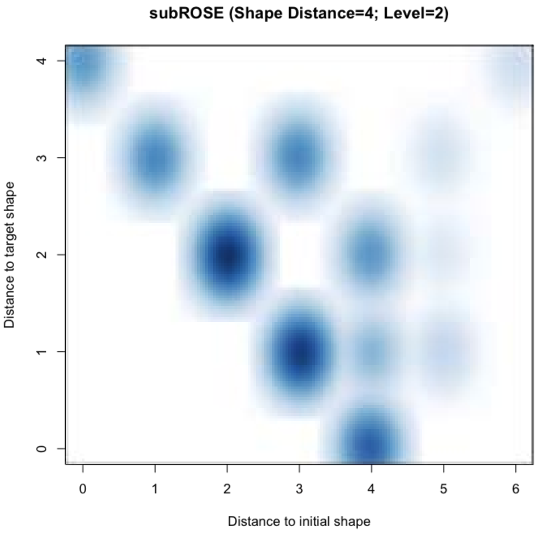

Figure 5.16: subROSE - Structure of optimal paths (Shape Distance = 4, Level = 2)

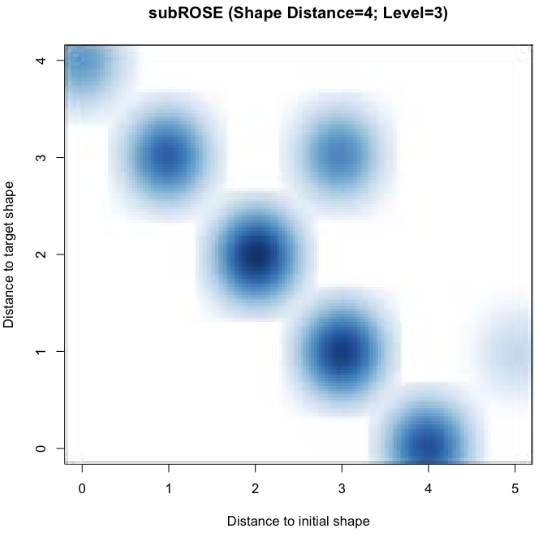

Figure 5.17: subROSE - Structure of optimal paths (Shape Distance = 4, Level = 3)

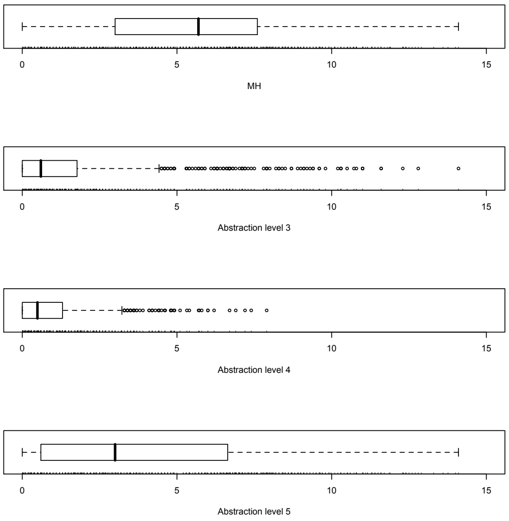

Figure 5.18: subROSE - Shape Network Approach (MH)

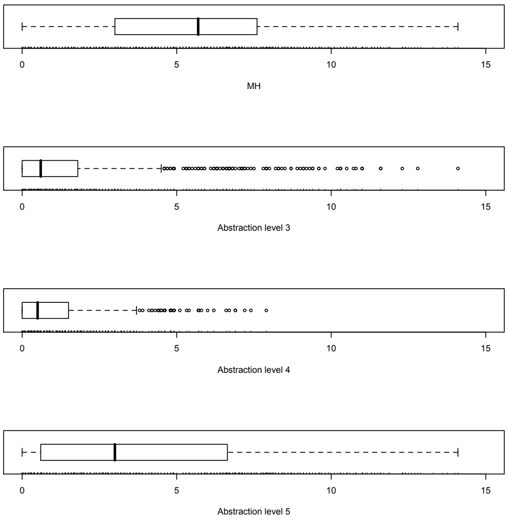

Figure 5.19: subROSE - Shape Tripples Approach (MH)

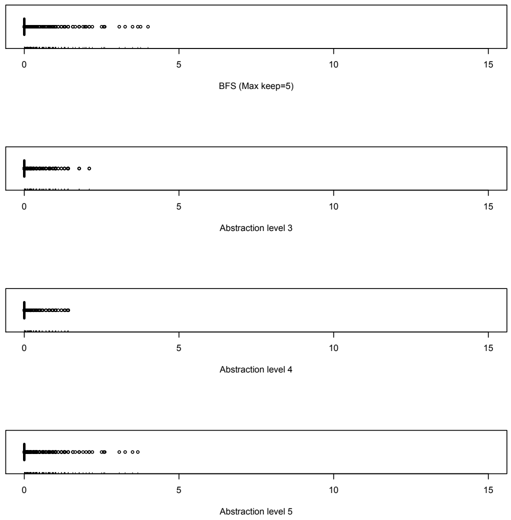

Figure 5.20: subROSE - Shape Network Approach (BFS, MaxKeep=5)

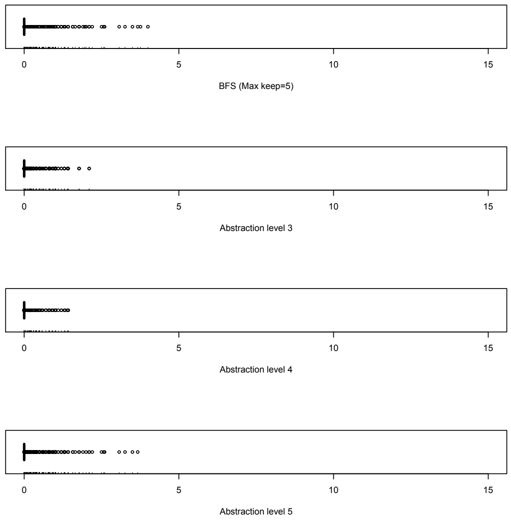

Figure 5.21: subROSE - Shape Tripples Approach (BFS, MaxKeep=5)

5.4.3 Shape Triples Approach

In Shape Network approach we consider all possible paths over shreps. But when we have a look at the result paths we can notice that the many of them have only one intermediate shrep. Thus it is interesting to consider whether using only one intermediate shrep how much precision we will lose. In Figure 5.19 the results of applying Shapes Triples approach (see Section 4.4) in combination with MH heuristics onto *subROSE* sequence are considered. Figure 5.21 shows the results of applying Shapes Triples approach in combination with BFS (Maxkeep=5) onto *subROSE* sequence. The results are very similar to the those of Shapes Network approach. Thus we can always use Shape Triples Approach instead of Shapes Network approach.

Next we consider a modification of Shape Tripples approach in which we take into account not only the shrep but a set of shape representatives.

In Figure 5.22 the results of applying Shapes Triples approach with sets (Size=5) in combination with MH heuristics onto *subROSE* sequence are presented. Second, Figure 5.23 presents the results of applying Shapes Triples approach with sets (Size=5) in combination with BFS ($MaxKeep = 5$) onto *subROSE* sequence. From these plots we can conclude that we do not gain more precision considering sets of representative structures for each shape (compare Figure 5.22 with Figure 5.19 and Figure 5.23 with Figure 5.21). We can explain this with the fact that the structures in the shape have similar structure (in particular several structures with the smallest energy). Thus it worth considering only the best representative.

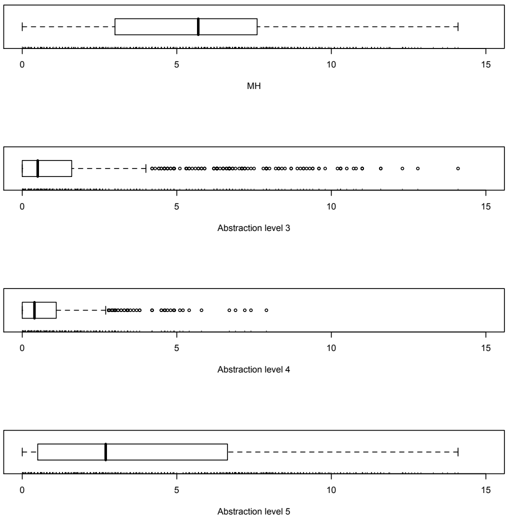

Figure 5.22: subROSE - Shape Tripples Approach (MH) with sets (Size=5)

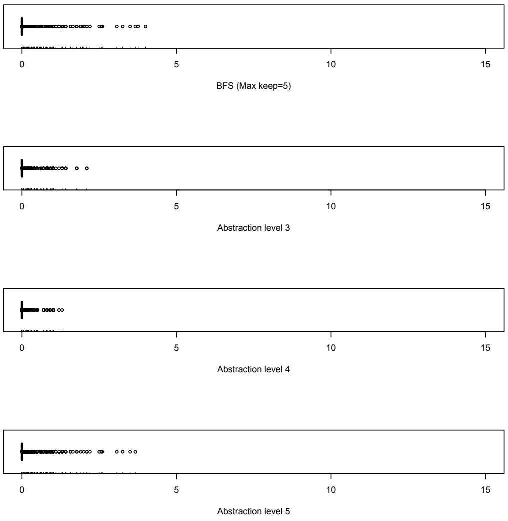

Figure 5.23: subROSE - Shape Tripples Approach (BFS, Maxkeep=5) with sets (Size=5)

5.4.4 Direct Shape Paths

The disadvantage of the previous approach is the need to look through the whole list of shapes. To tackle this problem we consider the next method: Direct Shape Paths approach. In Figure 5.24 the results of applying Direct Shape Paths approach in combination with MH heuristics onto *subROSE* sequence are presented. Figure 5.25 presents the results of applying Direct Shape Paths approach in combination with BFS ($MaxKeep = 5$) onto *subROSE* sequence. The structures of longer RNA sequences will have larger shape distance. Thus we can expect that we will get better results for *Caenorhabditis brenneri*. Figures 5.26 and 5.27 present results of Direct Shape Paths approach in combination with MH and BFS respectively. These figures agree with the above mentioned suggestion. We also can see that we get worse results then using Shape Triple Approach. This can be explained due to the fact that in some cases to get smaller barrier we need to consider a shape which is not on direct shape path. As in Shapes Network approach the results of Direct Shape Path with finer level of abstraction are not in general better then in case of coarser level of abstraction (as stated previously because of the non-monotonicity of shape abstraction).

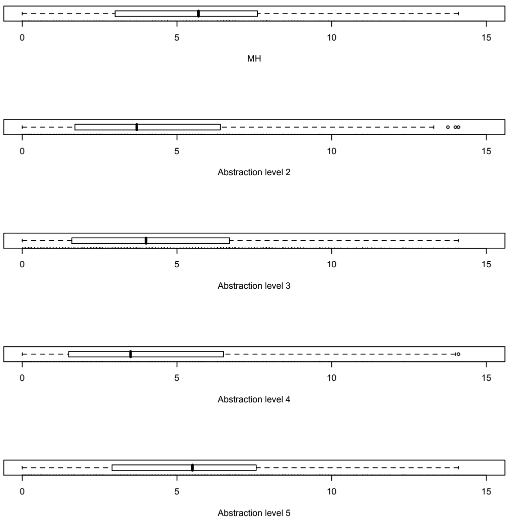

Figure 5.24: subROSE - Direct Shape Paths Approach (MH)

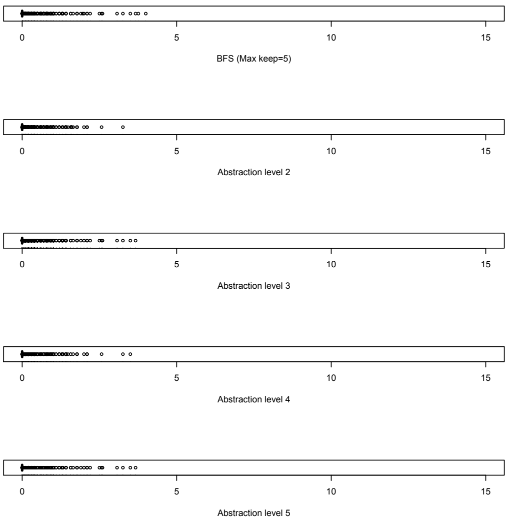

Figure 5.25: subROSE - Direct Shape Paths Approach (BFS, MaxKeep=5)

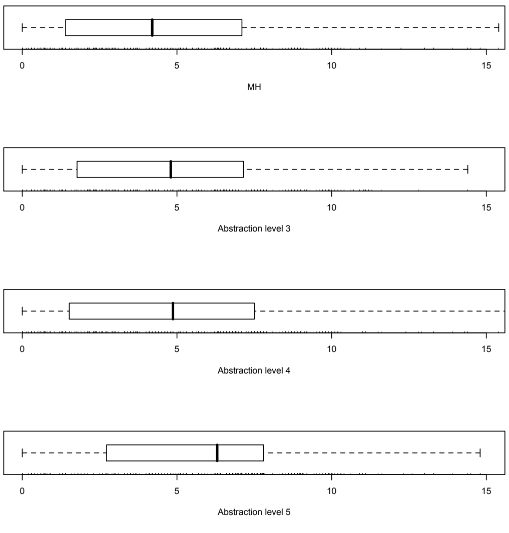

Figure 5.26: Caenorhabditis brenneri - Direct Shape Paths Approach (MH)

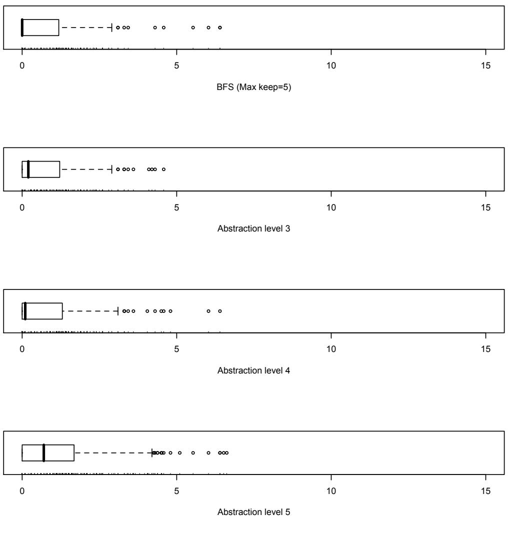

Figure 5.27: Caenorhabditis brenneri - Direct Shape Paths Approach (BFS, MaxKeep=5)

Chapter 6

Conclusions and Discussion

6.1 Conclusions

In this master thesis different methods for estimating barriers between RNA structures have been developed and compared to already existing ones. The approach of considering all possible direct paths is quite accurate but very time-consuming. In real world applications there are two algorithms which are mainly used: Morgan-Higgs heuristic and Breadth First Search. Both the methods distribute structures into classes and afterwards conduct search in the space of structures on the direct path. In order to get better results one can

1. introduce another ordering of structures in the class,
2. systematically consider paths which go somehow out of the direct path,
3. consider another distribution into classes.

First, we considered the first possibility. To introduce another ordering of structures in the class we used mountain metric (see Section 2.3). We have found out that we get the best results when we take into consideration the partial barrier and do not consider any information about mountain distance to the target structure (see Section 5.3). This shows us that unfortunately the mountain metric is inappropriate for the purpose of finding barriers.

Afterwards we analyzed the structure of optimal paths (see Section 5.2) . We found out that there are a lot of paths which are far from direct path. In order to tackle this problem we considered the shape abstraction (see Section 2.4). Two methods which use shape abstraction were first developed: shape network (see Section 4.3) and shape triples approach (see Section 4.4). Both methods have shown good results and scalability (see Section 5.4). We considered the question whether optimal paths are on direct shape paths (see Section 5.4.1) and found out that it is worth considering direct shape paths. This lets us make the search space smaller. A method called direct shape approach (see Section 4.5) was developed which uses this idea and conducts the search in the space of direct shape path. To summarize, the underlying idea of all the methods is the search for good intermediate points on the path which alloys us to consider not only direct paths and thus improve the quality of results. Second, the use of intermediate points let us apply MH and BFS on shorter distances and thus we can expect to get better intermediate results.

We showed that all the methods based on shape abstraction give better results then BFS and MH.

6.2 Future Work

We would like to point out tree directions of further research:

1. As we have seen both the direct shape approach and shape network approach crucially depend on the choice of good intermediate structures. In the future we would like to consider other abstractions which can lead to better results.

2. It would be useful to develop a criterion for choosing good intermediate point in shape triples approach.

3. All the presented algorithms need to be optimized in the future. In this way we can get both precise and efficient algorithms.

List of Figures

List of Algorithms

Bibliography

Baumstark, T., Schröder, A. & Riesner, D. (1997). Viroid processing: switch from cleavage to ligation is driven by a change from a tetraloop to a loop E conformation. *The EMBO Journal*, **16**, 599–610.

Chowdhury, S., Maris, C., Allain, F. & Narberhaus, F. (2006). Molecular basis for temperature sensing by an RNA thermometer. *The EMBO Journal*, **25**, 2487–2497.

Flamm, C., Fontana, W., Hofacker, I. & Schuster, P. (2000). RNA folding at elementary step resolution. *RNA*, **6**, 325–338.

Flamm, C., Hofacker, I., Maurer-Stroh, S., Stadler, P. & Zehl, M. (2001). Design of multistable RNA molecules. *RNA*, **7**, 254–265.

Flamm, C., Hofacker, I., Stadler, P. & Wolfinger, M. (2002). Barrier Trees of Degenerate Landscapes. *Zeitschrift für Physikalische Chemie*, **216**, 155–173.

Floyd, R.W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, **5**, 345.

Geis, M., Flamm, C., Wolfinger, M.T., Tanzer, A., Hofacker, I.L., Middendorf, M., Mandl, C., Stadler, P.F. & Thurner, C. (2008). Folding kinetics of large RNAs. *Journal of Molecular Biology*, **379**, 160–173.

Giegerich, R., Voss, B. & Rehmsmeier, M. (2004). Abstract shapes of RNA. *Nucleic Acids Research*, **32**, 4843–4851.

Hogeweg, P. & Hesper, B. (1984). Energy directed folding of RNA sequences. *Nucleic Acids Research*, **12**, 67–74.

Kochniss, H. (2008). *Ein Hybdridkinetik Ansatz fuer RNA Faltungswahrscheinlichkeiten*. Diplomarbeit, Friedrich Schiller University Jena.

Kubota, M. & Hagiya, M. (2005). Minimum basin algorithm: An effective analysis technique for dna energy landscapes. *Lecture Notes in Computer Science*, **3384**, 202–214.

Mann, M., Will, S. & Backofen, R. (2007). The Energy Landscape Library–a platform for generic algorithms. *Proc. of BIRD*, **7**, 83–86.

Morgan, S. & Higgs, P. (1998). Barrier heights between ground states in a model of RNA secondary structure. *Journal of Physics A: Mathematical and General*, **31**, 3153–3170.

Moulton, V., Zuker, M., Steel, M., Pointon, R. & Penny, D. (2000). Metrics on RNA secondary structures. *Journal of Computational Biology*, **7**, 277–292.

Perrotta, A. & Been, M. (1998). A toggle duplex in hepatitis delta virus self-cleaving RNA that stabilizes an inactive and a salt-dependent pro-active ribozyme conformation. *Journal of molecular biology*, **279**, 361–373.

Reeder, J. & Giegerich, R. (2005). Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, **21**, 3516–3523.

Richter, A.S. (2007). *Exploration of biopolymer energy landscapes via random sampling*. Diplomarbeit, Friedrich Schiller University Jena.

Stadler, P. (2002). Fitness landscapes. In *Lecture Notes in Physics*, 183–204, Springer.

Steffen, P., Voss, B., Rehmsmeier, M., Reeder, J. & Giegerich, R. (2006). RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, **22**, 500–503.

Steffen, P., Voß, B., Rehmsmeier, M., Reeder, J. & Giegerich, R. (2008). *RNAshapes 2.1.5 manual*.

Stillinger, F. & Head-Gordon, T. (1995). Collective aspects of protein folding illustrated by a toy model. *Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics)*, **52**, 2872–2877.

Team, R.D.C. (2004). *R: A language and environment for statistical computing*.

Ten Dam, E., Pleij, K. & Draper, D. (1992). Structural and functional aspects of RNA pseudoknots. *Biochemistry*, **31**, 11665–11676.

Uejima, H. & Hagiya, M. (2004). Analyzing Secondary Structure Transition Paths of DNA/RNA Molecules. *Lecture Notes in Computer Science*, 86–90.

Viennot, G. & De Chaumont, M. (1983). Enumeration of RNA secondary structures by complexity. *Mathematics in Biology and Medicine*, **57**, 360–365.

Wolfinger, M., Svrcek-Seiler, W., Flamm, C., Hofacker, I. & Stadler, P. (2004). Efficient computation of RNA folding dynamics. *Journal of Physics A Mathematical and General*, **37**, 4731–4741.

Wright, S. (1932). The Roles of Mutation. In *Inbreeding, Crossbreeding, and Selection in Evolution," in Proceedings of the Sixth Congress on Genetics*, 365.

Wuchty, S., Fontana, W., Hofacker, I. & Schuster, P. (1999). Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.

Zamora, H., Luce, R. & Biebricher, C. (1995). Design of Artificial Short-Chained RNA Species That Are Replicated by Q. beta. Replicase. *Biochemistry*, **34**, 1261–1266.

Zuker, M. & Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, **9**, 133–148.