Master Thesis

A Fragment Based Approach to RNA Threading

Christian Schudoma
(3680 750)

Dr. Frank Cordes
Zuse Institute Berlin (ZIB)
Dept. Numerical Analysis and Modeling
Computational Drug Design Group

Prof. Dr. Knut Reinert
Free University of Berlin (FUB)
Dept. Mathematics and Computer Sciences
Algorithmic Bioinformatics Group

Dipl.-Biol. Patrick May
Zuse Institute Berlin (ZIB)
Dept. Computer Science Research
Abstract

The analysis of the relations between structure and function of a biopolymer such as RNA is a core issue in structural biology and bioinformatics. The folding process of RNA structure follows a strict hierarchical scheme. The native structure of an RNA molecule is formed by the subsequent formation of secondary and tertiary structure motifs. These motifs are common in RNA molecules of different species, suggesting that parts (fragments) of a molecule with known structure can be used to predict unknown RNA structures. This thesis describes methods for an approach on the modeling of RNA three-dimensional structure based on structure fragments. We describe the design and composition of a novel template fragment library, containing about 25,000 template fragments from 578 different source structures. Based on this library, we semi-automatically model selected RNA structures, achieving 70% coverage of the native structure of an RNAse P A-type S domain. In addition, we present implementational details for a semi-automatic modeling framework employing an RNA sequence-structure alignment and an alignment validation method via isostericity matrices. The framework is tested on sample hairpin and tRNA targets. The results of this thesis demonstrate that building a target structure based on structure fragments from different template structures is a feasible approach for the RNA structure modeling and prediction problem.
Acknowledgements

This thesis has been written at the Computer Science Research (CSR) and Visualization and Data Analysis departments of the Zuse Institute Berlin. I want to express my thanks to the following people who have supported me during the months of work on the thesis:

I thank my mentor Dr. Frank Cordes (Zuse Institute Berlin) and Prof. Dr. Knut Reinert (FU Berlin) for supervising the thesis. Patrick May has invested a lot of his time into supporting me with the work of this thesis and provided me with a lot of references and materials including the miRNA stem finder tool. His office door and inbox was always open for my questions. Special thanks go to him for the exceptional support and for the fruitful discussions during long meetings. Special thanks go to Frank Cordes as well for his support and invested time. I thank Hon.-Prof. Hans-Christian Hege (Head of the Visualization and Data Analysis Department) and Dr. Thomas Steinke (Head of the BCB Junior Research Group, CSR Department) for making it possible for me to work on the thesis at their respective departments. I also thank Johannes Schmidt-Ehrenberg and Daniel Baum for their support during my time at the Visualization and Data Analysis department. Thanks to Dr. Gunnar Klau and Markus Bauer for providing the Lara algorithm. Ole Schulz-Trieglaff and Markus Bauer have proofread the final draft of this thesis. I thank them for their helpful hints and annotations.

Very special thanks go to my girlfriend Süreç Özcan not only for the long time we are together now (143L) but also for proofreading one of the earlier drafts and for accepting and understanding the endless hours I spent working on this thesis. Special thanks also go to my parents for their love and support of long years. Finally, I thank my friends-in-two-shifts for their moral support and for the wonderful times: Ben, Corinna, Gönenç, Steve, and Marcus during daytime and James, Sonia, Mariella, Robbie, and Karli (@) during nighttime.
## Contents

1 Introduction ............................................................. 1
1.1 Biological Background ............................................. 3
   1.1.1 Ribonucleic Acid ........................................... 3
   1.1.2 RNA Structure Organisation ................................ 5
1.2 Previous and Supportive Work in RNA Structure Modeling ........... 10

2 The Template Fragment Library ..................................... 13
2.1 Collecting Template Data .......................................... 13
2.2 The Template Fragment Format .................................... 14
2.3 Preprocessing Template Data ...................................... 15
2.4 Annotating Template Candidates .................................. 16
   2.4.1 Base-Pair Detection Using RNAVIEW ....................... 16
   2.4.2 Stem Detection ............................................... 18
   2.4.3 Secondary Structure Motif Assembly ....................... 20
   2.4.4 Region Identification ....................................... 22
   2.4.5 Tertiary Structure Motif Assembly ....................... 22

3 RNA Threading Methods and Algorithms ............................ 25
3.1 The Threading Framework ......................................... 25
3.2 Lagrangian-Relaxed RNA Structure Alignment (Lara) ............... 27
3.3 Hairpin Detection .................................................. 32
3.4 Contact Validation (Isostericity Matrices) .......................... 34
3.5 Fragment Assembly via Clique Detection ........................... 35

4 Computational Results ................................................ 41
4.1 The Template Fragment Library ................................... 41
   4.1.1 Library Creation ............................................ 41
   4.1.2 Library Statistics ........................................... 42
4.2 Testing of Framework Components .................................. 45
## CONTENTS

4.2.1 Lara Alignment Validation ........................................... 45  
4.2.2 Isostericity Matrix Validation ...................................... 49  
4.3 Semi-automatic modeling of selected structures .......................... 52  
4.3.1 Proof of concept: tRNA ........................................... 53  
4.3.2 Proof of concept: A-Type S Domain of RNase P ....................... 56  
4.4 RNA Design: Hairpin Modeling .......................................... 60  
4.4.1 Modeling of the P12 Hairpin of the A-Type S Domain of RNase P ........ 60  
4.4.2 Modeling of a HIV-1 A-rich Hairpin Loop ............................ 62  
4.5 Modeling of tRNA Structures ............................................. 63  

5 Conclusion ................................................................. 67  
5.1 Problem Discussion ...................................................... 67  
5.1.1 The Template Library ............................................... 68  
5.1.2 RNA Structure Alignment ........................................... 69  
5.2 Outlook ................................................................. 69  
5.2.1 Extension and Refinement of the Template Library .................... 69  
5.2.2 Selecting Template Candidates for RNA Design ....................... 69  
5.2.3 Isostericity Matrices ............................................... 70  
5.2.4 Fragment Assembly ............................................... 70  

A Definitions ............................................................... 73  

B Library Contents ......................................................... 75  

C Parameters ............................................................... 79  
C.1 Hairpin Detection ..................................................... 79  
C.2 Lara ................................................................. 79
List of Tables

1.1 Leontis/Westhoff Base-Pair Families. ........................................ 6
4.1 Semi-automatic Modeling of 2tra: Hairpins ................................. 54
4.2 Semi-automatic Modeling of 2tra: Kissing Hairpins ...................... 55
4.3 Semi-automatic Modeling of 2tra: Multiloops .............................. 55
4.4 Semi-automatic modeling of 1u9s: Template Fragments .................. 58
4.5 Modeling of tRNA Structures: Template Fragments ...................... 64
List of Figures

1.1 The Ribonucleic Acid Backbone ............................................ 3
1.2 Hydrogen Bond Interaction Sites ......................................... 4
1.3 cis/trans Base-Pair Configurations ..................................... 5
1.4 Isostericity Matrices for the Twelve Base-pair Families .......... 6
1.5 The three levels of RNA structure ........................................ 7
1.6 Secondary Structure Motifs .............................................. 8
1.7 Tertiary Structure Motifs: Kissing Hairpins ...................... 9

2.1 Template Library Creation Data Flow ................................. 13
2.2 Example Contact Map for 2tra ......................................... 14
2.3 Structure Annotation Data Flow ....................................... 16
2.4 Sample Base-Pair Coordinate Systems ............................... 17
2.5 Stem Detection Cases .................................................... 19
2.6 An Example RNA Secondary Structure ......................... 21
2.7 Sample Output of the Secondary Structure Motif Assembly .... 21
2.8 Subsequence Regions for Different Secondary Structure Motif Types .... 22

3.1 The RNA Threading Framework ........................................ 26
3.2 A Scheme of RNA Secondary Structure ............................ 27
3.3 An Example RSA Graph .................................................. 28
3.4 Alignment Scattering ..................................................... 33
3.5 Hairpin Detection by Local Alignment ............................... 33
3.6 Fragment Assembly: Anchor Points and Overlaps .............. 36
3.7 Fragment Overlap Types ............................................... 37
3.8 Graph-Based Fragment Assembly ................................... 39

4.1 Fragment Library Data: Distribution of RNA Species - Source Molecules .... 42
4.2 Fragment Library Data: Distribution of RNA Species - Fragments .......... 44
4.3 Fragment Library Data: Distribution of Fragment Types ........ 44
4.4 Test Alignments for 2tra .............................................. 46
LIST OF FIGURES

4.5 Dotplots: Predicted and Verified Secondary Structures for 2tra .................. 48
4.6 tRNA Test Alignments for 2tra .................................................. 48
4.7 Gapless Threading of 2tra - Realised Contacts ........................................ 51
4.8 Gapless Threading of 2tra - Unaligned Contacts ....................................... 51
4.9 2tra Reversed Gapless Threading: Aligned Contact Map ................................ 52
4.10 Assembly of a tRNA Model Structure ................................................... 54
4.11 Distribution of RNA Species among Hairpin Source Structures .................... 56
4.12 Native Structure for the A-Type S Domain of RNAse P .............................. 58
4.13 Model Structure for the A-Type S Domain of RNAse P .............................. 59
4.14 Superposition of the P12 Hairpin with Template Hairpins .......................... 61
4.15 Superposition of an HIV-1 A-rich Hairpin with Template Hairpins ................. 62
4.16 Model Structure for 2tra ................................................................. 65
4.17 Model Structure for 6tna ................................................................. 65
4.18 Model Structure for 1j1u ................................................................. 66
Chapter 1

Introduction

The biological function of a biopolymer is strongly influenced by its three-dimensional structure. The analysis of the relations between structure and function as well as the exploration of the underlying complex folding mechanisms are core issues in structural biology and bioinformatics. The accurate elucidation of molecular structures is of high importance for the comprehension of these relations. Structure elucidation experiments from the traditional fields of X-ray crystallography and nuclear magnetic resonance spectroscopy yield a vast quantity of macromolecular 3D structures with each passing month. As these experiments are costly and time-consuming, the computational prediction and modeling of three-dimensional structures is of major interest in structural bioinformatics.

While the majority of the research of the field focuses on protein structure, the structural exploration of nucleic acids has only been emergent for the last decade [18]. With the exception of two transfer-RNA (tRNA) structures determined in the early 1970s, only very little to nothing was known about the actual three-dimensional structures adapted by the different RNA species until the discovery of the structure of hammerhead ribozymes in 1994 [12]. This former lack of interest in RNA structure can be explained by the fact that prior to the discovery of the catalytic \textit{(ribozymatic)} properties of ribosomal ribonucleic acids (rRNA) by Cech and Altman in 1982/1983 [14, 18, 20, 28] the at that time known RNA species (tRNA, rRNA, messenger-RNA (mRNA), and viral RNA (vRNA) ) had been disregarded as simple information carriers of mostly linear structure (mRNA) within the process of protein biosynthesis (a function in which three dimensional structure was believed to play (if at all) just a minor role). In addition, the rather complicated crystallisation of RNA molecules has always been a limiting factor for the classical structure determination via X-ray crystallography: the majority of RNA crystals are not large enough or do not diffract X-rays to yield biochemically informative structures [17, 41]. Also, the possibility of crystallising a catalytically irrelevant conformation is higher for RNA molecules than for protein molecules, due to RNA enzymes being much more dynamic than proteins [19].
Despite the fact that the structure of most RNA molecules is in general less complex than the fold architecture of most proteins, the underlying folding mechanisms are of comparable complexity. The formation of the native structure of an RNA molecule follows a strict hierarchy: starting with the primary structure – the sequence of covalently bound ribonucleotides ("bases") – folding into an initial energetically favourable secondary structure by formation of hydrogen bonds between individual bases. The tertiary structure is then formed by hydrogen bond formation between secondary structure elements, folding the molecule into its native three dimensional structure [4]. For the folding process, the presence of positively charged metal ions such as $Mg^{2+}$ is essential [54].

The perception of RNA being only an information carrier of simple and rather non-functional structure has been proven wrong over the last decades (cf. e.g. [18]). The different RNA species carry out a wide and versatile range of essential biological functions, often fulfilling tasks which have been believed as being guided by proteins in the past. For instance, peptidyl transfer – linkage of amino acids to a polypeptide chain during protein biosynthesis – which was formerly believed to be performed by proteins is actually performed by ribosomal RNAs. Interactions between RNA and proteins or other (ribo-)nucleic acids regulate important cellular activities, such as gene expression or ribozymatic activity. The so-called RNA interference (RNAi) – the repression of gene expression by small double stranded RNA fragments (miRNA/siRNA) – is one of the more recently discovered functions of RNA. Research on RNAi is of high pharmaceutical interest as the gene-regulative effect of RNAi appears to offer new therapy approaches for yet uncurable diseases, such as diabetes type II, multiple sclerosis, or HIV infection [18, 44, 57]. Another class of RNA molecules, the aptamers, are capable of specifically binding to certain molecules showing high affinity towards their given target. Thus, aptamers exhibit properties that are typical for protein receptors or antibodies. While the term aptamer is usually used for an engineered molecule, they can occur in nature as part of riboswitches. These are RNA molecules controlling the expression of genes. A conformational change of the riboswitch-structure induced by ligands binding to its aptamer domain determines whether the riboswitch function is permitted ("switched on") or precluded ("switched off") [18].

The close connection between RNA function and three-dimensional structure suggests research targeting at RNA structure to gain deeper insight and understanding of the molecular function. Due to the hierarchical fold architecture – which builds up a complete RNA structure from scratch by subsequential formation of structure motifs – the application of a modeling approach based on structure fragments appears promising.
1.1 Biological Background

1.1.1 Ribonucleic Acid

Ribonucleic Acids (RNA) are biopolymers consisting of one or more molecular chains of ribonucleotide monomeric units. A ribonucleotide contains a ribose residue with a mono-phosphate residue ($PO_4$) which is bound to the ribose-$O_5^*$-atom via an $N$-glycosidic bond, and a nucleobase (purine or pyrimidine) covalently bound to the ribose-$C_1^*$-atom. The $P, O_5^*, C_5^*, C_4^*, C_3^*$, and $O_3^*$ atoms of the ribose and phosphate residues form the so-called backbone of a chain of ribonucleotides. Two succeeding ribonucleotides $r_1, r_2$ are connected through the formation of an $O$-glycosidic bond between the $O_5^*$-atom of $r_1$ and the $PO_3$ residue of $r_2$ (cf. Fig. 1.1). The purine and pyrimidine bases reflect the genetic information encoded in an RNA molecule and are crucial for the formation of its molecular structure.

![Figure 1.1: The Ribonucleic Acid Backbone](image)

Figure 1.1: The Ribonucleic Acid Backbone The figure displays the structure of the ribonucleic acid backbone. $r_1$, $r_2$ denote backbone parts, $s_1$, $s_2$ denote ribose-residues, and $b_1$, $b_2$ denote nucleobase-residues. The backbone part of a ribonucleotide includes its $P, O_5^*, C_5^*, C_4^*, C_3^*$, and $O_3^*$ atoms. Blue arrow: the complete backbone between the depicted ribonucleotides.

Purines and Pyrimidines

Analogous to DNA, there exist four different standard (nucleo-)bases contained in an RNA molecule: the purines adenine (A) and guanine (G) and the pyrimidines cytosine (C) and uracil (U).
(U). Uracil replaces thymine (T), which is contained in DNA, in most RNA molecules. However, some transfer-RNAs contain a small amount of thymine as well. Additionally, transfer-RNAs can contain a small number of converted adenine bases (hypoxanthine (I)).

Base-Pairs

Nucleobases contain hydrogen atoms which can establish hydrogen bonds with certain acceptor atoms of other nucleobases. Two bases that are connected by at least two hydrogen bonds are called a base-pair. For sterical reasons base-pair formation can only occur between bases that are three or more positions apart on the ribose-phosphate backbone. Hydrogen bond interactions can occur on three virtual edges of a nucleobase: the Watson-Crick edge, the Hoogsteen edge, and the Sugar edge (cf. Fig. 1.2). This means that a base can in theory form base-pairs with up to three other bases (one per edge). Additionally, pseudo-base-pairs can be formed by single bonds between either a base residue and a ribose residue or two base residues.

Figure 1.2: Hydrogen Bond Interaction Sites. Left: purine interaction sites. Right: pyrimidine interaction sites. W: Watson-Crick edge, H: Hoogsteen edge, S: sugar edge. Figure reproduced from [32].

The different types of base-pairs can be classified into twelve families, according to Leontis and Westhoff [32]. Base-pair families are identified by the interacting edges of the pairing bases together with the relative orientations of the base-ribose glycosidic bonds (cis or trans). Figure 1.3 exemplarily displays the cis/trans configurations of an A-U Watson-Crick (W/C) pair. A base-pair is said to be in cis- configuration if both glycosidic bonds are situated on the same side of a reference plane perpendicular to the line connecting the centroids of the base ring systems. Otherwise it is in trans- configuration. The twelve existing families are listed in Table 1.1. Within a base-pair family certain types of base-pairs can be isosteric to each other, i.e. the distances between the $C^\gamma$-atoms of both pairs are nearly the same and the glycosidic bonds are oriented in the same way. Due to these properties, isosteric base-pairs occupy a highly similar space and thus can be exchanged against each other without disturbing the overall three-dimensional structure (i.e. the path and relative geometric orientation of the ribose-phosphate backbone) of the molecule. For instance, the cis W/C A-U and G-C base-pairs are isosteric to each other, while the cis W/C G-U wobble pair is neither isosteric to the A-U nor to the G-C pair. A classification of the isostericity of the different base-pair families (isostericity matrices) has been published by
1.1. BIOLOGICAL BACKGROUND

Leontis, Stombaugh, and Westhoff [31] and Lescoute et al. [34]. Figure 1.4 depicts the isostericity matrices for the 12 different base-pair families. Note that for most families the matrices are not completely filled. The missing base-pairs can either not be formed due to sterical reasons (e.g. the cis W/C G-G pair) or had not been observed when the classification was defined.

Figure 1.3: cis/trans Base-Pair Configurations. Configurations are shown exemplary for an A-U Watson-Crick pair. Top: cis-configuration, Bottom: trans-configuration. Arrows pointing in the direction from the pyrimidine/purine N9/N1 atoms to the ribose-C1* atom. The reference planes are depicted as grey lines. Figure reproduced from [32].

1.1.2 RNA Structure Organisation

The native structure of RNA molecules can be divided into three different levels of organisation: primary, secondary, and tertiary structure. Figure 1.5 illustrates the three different levels of RNA structure on an example tRNA molecule (PDB:2tra).

The primary structure denotes the ribonucleotide sequence (commonly referred to as base-sequence) of the molecule. Usually, the base-sequence of an RNA molecule only consists of a combination of the bases A, G, C, U. Thymine and hypoxanthine are analogous to uracil (or adenine, respectively) and thus represented by their corresponding standard base. Furthermore, modified bases such as pseudouracil (Ψ) are represented by their most-similar standard base.

The secondary structure is formed by a subset of the cis- Watson-Crick/Watson-Crick base-pairs contained in an RNA molecule. This includes the standard (canonical) A-U and G-C pairings already known from the formation of DNA helices as well as the so-called G-U wobble-pairs. Successive base-pairs form energetically favourable and thus stable stem-regions. The unpaired re-
CHAPTER 1. INTRODUCTION

Table 1.1: Leontis/Westhoff Base-Pair Families. Families that differ only by their cis/trans-orientations are combined in the same row: family $x$ of $x/y$ is in cis orientation, family $y$ is in trans orientation. a: anti-parallel local strand orientation, p: parallel local strand orientation.

Figure 1.4: Isostericity Matrices for the Twelve Base-pair Families. The figure displays the isostericity matrices for the twelve known base-pair families as classified by Leontis and Westhoff [31]. The matrices are to be read from 5’ to 3’, i.e. the isostericity class of a base-pair $(b_i, b_j)$ is located in the $i$-th row and the $j$-th column. Different colours denote the different isostericity classes. Isostericity classes are only defined within their base-pair family. Note that two base-pairs belonging to different families cannot be isosteric. Figure reproduced and modified from [31].
1.1. BIOLOGICAL BACKGROUND

Figure 1.5: The three levels of RNA structure. a) Primary structure (base sequence) of 2tra. b) Secondary structure graph of 2tra. c) Three-dimensional structure of 2tra representing the effects of tertiary contacts within the molecule. Note the two imposed D- and T-loops at the upper right part of the molecule which together with the coaxial stacking of the respective stem-regions induce the overall three-dimensional L-shape of the molecule.
regions between two stems are called loops. The secondary structure of an RNA molecule is formed by a number of secondary structure segments (motifs). Such a secondary structure motif consists of exactly one incoming stem region (mother-stem), a loop region, and an optional number of outgoing stems (child-stems). Two adjacent motifs \( n_i, n_{i+1} \) are overlapping: the mother-stem of \( n_{i+1} \) is equivalent to one of the child-stems of \( n_i \). Secondary structure motifs can be classified into four regular loop classes (bulge, hairpin, internal loop, and multiloop) which do not interfere with the three-dimensional structure of the molecule (cf. Fig. 1.6). A fifth loop class, the pseudoknot creates a change of the threedimensional structure. The four regular loop classes can be divided into two groups: bulge, internal loops, and multiloops belong to the group of the interior loops while the group of external loops only consists of hairpin loops. The occurrence of an interior loop induces either a junction (multiloop) or elongation (internal loop, bulge) in the current part of the structure. An interior loop always ends with one or more "open" stem regions, as the base-pairs of a stem region are not connected by the ribose-phosphate backbone. Hence, a sequence of interior loops has always to be finalised by a number of external loops equivalent to the number of child stems of the outermost interior loop. Two secondary structure motifs are said to belong to the same fold class if they are structurally identical, e.g. two 3-multiloops with identical mother-stem, child-stem and unpaired-region sizes belong to the same fold class. Note that the sequences of two motifs belonging to the same fold class may differ. The topology given by the secondary structure motif architecture is maintained in the overall structure fold of the molecule. E.g., the typical secondary structure of a tRNA consists of a 3-multiloop with three outgoing hairpin loops. This secondary structure is commonly referred to as cloverleaf or butterfly (cf. Fig. 1.5b).

![Figure 1.6: Secondary Structure Motifs.](image)

Base-pairs that do not belong to the secondary structure together with pseudo-base-pairs form the tertiary structure of the molecule. This includes other atomic interactions such as van der Waals forces, electrostatic and hydrophobic interactions and hydrogen-bonds between e.g. base and ribose residues. Tertiary contacts are interactions between distinct secondary structure elements. They induce local and/or global structure folds and as such are dominantly responsible for the overall three-dimensional structure of an RNA molecule [4].
Tertiary Structure Motifs

Tertiary interactions can occur between two helical motifs (stem-stem), between two unpaired regions (loop-loop), and between an unpaired region and a stem region (loop-stem) [4]. In the three-dimensional structure of a tRNA molecule (cf. Figure 1.5), the stems of the D-loop and the T-loop, as well as the acceptor-stem and the stem of the anticodon-loop stack upon another (coaxial stacking stem-stem interaction). The typical L-shape of a tRNA molecule is yielded by the stacked stem regions as well as the kissing hairpin loop-loop interaction between the D-loop and T-loop hairpins. An example for a kissing hairpin motif is depicted in Figure 1.7 which displays the interaction between the D-loop and T-loop of the tRNA 2tra. Coaxial stacking is shown in Figure 1.5c). For a more detailed overview of the different tertiary motifs, the reader is referred to the works of Batey et al. [4] and Leontis and Westhoff [33].

The Hierarchical Fold Architecture of RNA

The native three-dimensional structure of an RNA molecule is regarded as being strictly hierarchical. Small secondary structure elements (stems and single hairpins) already start forming during the synthesis of its primary structure, i.e. the formation of the ribose-sugar backbone by RNA Polymerases. Successively, most of the larger secondary structure elements are formed. Starting from this so-called unfolded state the sequential formation of tertiary contacts and remaining secondary structure motifs occurs, eventually yielding the native structure of the RNA molecule. Hence, the RNA folding process differs from protein folding, where tertiary interactions are only formed after the complete secondary structures have been established. Tertiary interactions can be seen as playing a main role in the folding of RNA molecules into their native three-dimensional
1.2 Previous and Supportive Work in RNA Structure Modeling

Approaches that target at modeling an RNA structure from a target nucleotide sequence or supporting the actual structure modeling process can be classified into certain categories.

*Ab initio* methods attempt to simulate the natural folding process of a target sequence. These methods utilise predefined physical parameters together with energy parameters (*force fields*) resulting from physicochemical experiments. They exist both on the *contact-level* trying to predict base-pair formation purely from sequence data, and on the *physical level* attempting to (re-)construct the actual formations of atomic bonds and complete residues from the ground up. A major drawback of these methods are the errors originating from the force field parameters, e.g. by the interpolation or approximation of energy values. Also, ab initio methods traditionally require a large amount of computing resources. Nussinov [42] and Zuker [58] have described ab initio methods for the prediction of RNA secondary structure. An approach for the prediction of pseudoknot structures has been proposed by e.g. Rivas and Eddy [46] and Witwer, Hofacker, and Stadler [55]. Due to the long running times, the approaches are only applicable to rather short sequences.

*Threading* approaches are fold-recognition methods [1] that attempt the modeling of molecular structures based on available structure data. They aim at finding a structure from a pool of experimentally verified *template* structures which best fits the target sequence. The target sequence is iteratively threaded through each template structure. The best fitting structure is then used as a template for assembling the structure of the target sequence. These approaches limit the search space to the conformations of known structures [29]. This can be seen as both an advantage (lower running time), and a disadvantage if the target’s native structure differs significantly from the template structure.

The *fragment-based* prediction of three-dimensional structures originates from the field of protein structure modeling. The *fragment assembly* approach (e.g. ROSETTA) [29, 50] utilises a library of known template structures. These templates serve as a source for small gapless structure fragments which show local sequence similarity to the target sequence. The initial structures resulting from the fragment search are then submitted to an optimisation method, such as Simulated Annealing (SA) for a final model structure assembly. As for the fragment-based modeling of RNA structures, Sykes and Levitt propose an approach utilising pairs of ribonucleotides (doublets). The two nucleotides of a doublet are not necessarily adjacent on the ribose-sugar backbone, thus allowing the coverage of secondary and tertiary structure base-pairs in addition to sequential
1.2. PREVIOUS AND SUPPORTIVE WORK IN RNA STRUCTURE MODELING

(Semi-)Automatic Modeling methods utilise verified or experimental structure data such as atomic distances, atom coordinates, or even a complete structural graph to assemble an actual three-dimensional model. Examples for semi-automatic modeling approaches are intelligent systems (IS) e.g. MC-SYM [35] or YAMMP [53]. These approaches do not provide any kind of prediction but might contain a force-field [10, 48] relaxation method to correct bond lengths and angles.

The sequence and/or structure alignments from the field of classical sequence analysis, e.g. by Needleman and Wunsch (global alignment) and Smith and Waterman (local alignment) [40, 51] and their extension to incorporate secondary structure information by Sankoff [47], Lenhof, Reinert, and Vingron [30], or Bauer and Klau [6, 7] can be used as residue mappings which serve as a guideline for (semi-)automatic modeling tools. These methods can give insight in conserved subsequences and/or substructures.

The field of RNA threading is relatively small. Hence, there exists only a very limited number of similar previous approaches. The immediate precursor to this thesis is the RNA threading approach by Boit [10, 11], who employed a genetic algorithm operating on a template library consisting entirely of a very small number of complete tRNA structures. While the approach has yielded quite promising results regarding the modeling of tRNA structures the drawbacks of using complete template structures are obvious: the overall quality of a structure model might decrease if certain regions do not fit onto the native structure of the target molecule. The restrictions on the contents of the template library do not allow the application of the approach to types of RNA other than tRNA. The approach was later altered to employ a combined RNA primary and secondary structure alignment [6, 7, 49].

This thesis presents a new threading approach for the modeling of the three-dimensional structure of RNA: the segment- or fragment-based architecture of RNA secondary structure theoretically allows the assembly of an RNA secondary structure using structure fragments of different RNAs. In contrast to the described fragment-based approach from protein structure prediction, the resulting template fragment library will not only consist of gapless structures. Instead, we attempt to expand the fragment-based assembly onto the tertiary structure level. The produced fragments comply to the motifs found in the secondary and tertiary structures of the used RNA template molecules. The applicability of the approach is proven by semi-automatically building example structures of a complete tRNA and of a domain from an RNAse P complex, using structure fragments from other RNA molecules.

We describe the design of a novel and versatile library of template structures containing about 25,000 structure fragments originating from 578 source molecules. In addition, we propose a semi-automatic modeling pipeline (framework) and provide details of its implementation. The
framework employs an RNA sequence-structure alignment [6, 7], detection of stem regions [36]), and ab initio secondary structure prediction [22]. The assembly of partial and complete RNA three-dimensional structures based on structure fragments of different RNA molecules is demonstrated on selected examples.

Chapter 1 includes an introduction to the biological background and an overview of RNA and general molecular structure modeling methods and previous works. It describes the general idea of the main topic of the current work: the fragment-based RNA threading approach.

In Chapter 2 we describe the design process and implementation of the template fragment library.

Chapter 3 contains a detailed characterisation of the RNA threading framework together with descriptions of the employed methods and algorithms: the Lagrangian-relaxed RNA structure alignment algorithm, hairpin search via miRNA signal detection, contact validation via isostericity matrices, and fragment assembly via clique detection.

Details about the performed experiments and their results are given in Chapter 4, including several tests of the different framework components.

Finally, Chapter 5 closes the thesis with an evaluation of the results and a problem discussion. An outlook to possible future work is given.

In the appendix section, additional information can be found: Appendix A contains formal definitions, Appendix B lists the contents of the fragment library, and Appendix C consists of the parameter settings for the employed algorithmic methods.
Chapter 2

The Template Fragment Library

This chapter describes the methods and algorithms used for the design and automatic creation of the template fragment library. Figure 2.1 displays the data flow for the creation of the library. The single steps are examined more closely in the corresponding sections.

Figure 2.1: Template Library Creation Data Flow. Structural data from the public SCOR and PDB databases provide an initial set of possible template candidates. The application of several length and content filters on the initial data set yields a number of feasible and valid template candidate structures. The set of template candidates is annotated by secondary structure detection routines resulting in the final template library.

2.1 Collecting Template Data

Databases covering nucleic acid structural data are sparsely distributed among the publicly available biomolecular structure resources on the world wide web. In fact, there exist only a few major sites, namely the Rutgers’ Nucleic Acid Database (NDB)[8], the Brookhaven Protein Data Bank (PDB)[9], and the Baltimore RNA Structure Database (RNABase)[38]. Of these three databases only the PDB features a repository of experimentally verified three-dimensional structures of different types of biomolecules including nucleic acids. In contrast, NDB and RNABase use references to PDB data for their database entries instead of featuring their own structure repositories.
As of May 2006, the PDB holds 987 entries containing RNA structure data, opposite to NDB (967 entries) and RNABase (914 entries). The 987 RNA-containing PDB entries form the data foundation for the threading template library. (cf. Figure 2.1)

The Berkeley Structural Classification of RNA database (SCOR)[25], was established to provide a similar functionality for nucleic acids as the Structural Classification of Proteins database (SCOP)[39] provides for proteins: the classification and annotation of common structural motifs found in different molecules. Unfortunately, the SCOR has not been updated since May 2004 and thus does not contain data for more recently deposited PDB entries. Still, the available data (579 entries) could 1) serve as a guide to structurally interesting PDB entries and 2) provide annotations for a subset of the data obtained from the PDB.

![Figure 2.2: Example Contact Map for 2tra](image)
The ribonucleotide sequence (primary structure) is annotated with secondary structure (solid lines) and tertiary structure contacts (dashed lines). Please note that not every single secondary structure contact is depicted individually. To simplify matters, the secondary structure stems are displayed as a whole.

### 2.2 The Template Fragment Format

A library entry (contact map) for a template-fragment consists of primary structure, secondary structure, base-pair information, and tertiary structure data (cf. Fig. 2.2). The primary structure of a fragment contains one or more continuous subsequences of the main sequence of the RNA molecule (cf. Sect. 2.4 - Region Identification). The secondary structure holds the information about all secondary structure stems contained in the fragment, while the tertiary structure consists of the tertiary contacts. The base-pair information contains classification data about all base-pairs (including tertiary contacts) contained in the fragment.
2.3 Preprocessing Template Data

Three different preprocessing tasks are performed on a PDB entry file. These tasks are chain filtering, sequence filtering and redesign, and sequence index normalisation. The PDB entry files that are not removed by the preprocessing routines form a set of feasible and valid candidates (source structures) for the template library.

1. **Chain Filtering.** Due to the difficulties in nucleic acid structure analysis (e.g. problems during crystallisation) compared to protein structure, structure elucidating experiments are usually not targeted at nucleic acids but at proteins instead. Therefore most experimentally verified nucleic acid structures are either by-products of structure verification experiments aiming at proteins forming molecular complexes with nucleic acids or they might have been crystallised as a complex, due to the better crystallisation properties of proteins. In practice this means that the majority of PDB entries for nucleic acid structures contain fragments from protein structures in addition to the actual nucleic acid data. These chains might induce local structural changes in the respective RNA structure. As only pure ribonucleic acid structures are to be included in the template library, chains consisting of protein, DNA, or other (bio-)polymers are removed.

2. **Sequence Filtering and Redesign.** A chain of an RNA molecule can contain non-nucleobase residues. This is the case for e.g. tRNAs to which an amino acid is bound or for engineered RNA structures including marker or replacement residues. Another case is a modified residue, e.g. Pseudouridine instead of Uridine. If such residues can be safely exchanged against a standard nucleobase, i.e. without risking a misclassification by the structure annotation, then the corresponding standard nucleobase will be annotated to them. Otherwise the chain will be removed. Terminal-bound non-nucleobase residues like the amino acid transported by tRNA molecules will simply be cut off instead of discarding the whole chain. shorter than a certain number of residues will be well. Stereochemical reasons prohibit the folding short chains into a secondary or tertiary these chains can be regarded as unimportant contribute a structure fold to the template.

3. **Sequence Index Normalisation.** To ensure smooth processing of template data by the annotation methods further downstream (cf. Fig. 2.1, Fig. 2.3) in the data flow the sequence numbering system used in a PDB entry has to be checked for continuity. Certain indices might be missing in an entry or additional residues might have been inserted to represent alternate conformations. In addition, the numbering systems of different entries do not necessarily start at a common index. To prevent errors due to a discontinuous sequence numbering and facilitate the application of the annotation methods, a normalised numbering system, sequentially enumerating the residues starting by 0, is used.
CHAPTER 2. THE TEMPLATE FRAGMENT LIBRARY

Figure 2.3: Structure Annotation Data Flow. A base-pair detection routine determines the secondary structure and tertiary contact information from the 3D atom coordinates of a candidate molecule. A graph representing the secondary structure of the molecule (structural graph) is then built up using both primary and secondary structure information. On this graph a stem detection routine is performed yielding an annotation of the base-pair stems contained in the molecule. Based on this annotation the complete secondary structure of the molecule is assembled. The region identification routine then decides which residues belong to a certain structure motif, eventually resulting in a set of secondary structure motifs. Using the tertiary contact data, the tertiary structure motif assembly routine identifies secondary structure fragments that are connected by tertiary interactions and thus forming a tertiary structure motif.

2.4 Annotating Template Candidates

This section describes the methods and algorithms used to annotate the template candidate molecules (cf. Fig. 2.3). Candidates that cannot be annotated are removed during the annotation process. Successfully annotated candidates yield a number of template fragments corresponding to the annotated secondary and tertiary structure motifs, forming the final template fragment library.

2.4.1 Base-Pair Detection Using RNAVIEW

To obtain information about the base-pairs contained in a RNA molecule, we employ the tool RNAVIEW [56] by Yang et al.. RNAVIEW detects possible base-pairs by pairwise comparison of the vertical distances and angles between base-planes. Given a set of atom coordinates (in PDB file format) of an RNA molecule RNAVIEW generates a base-pair annotation. We can then use this annotation to reconstruct the secondary-structure of the RNA molecule. The theory behind RNAVIEW is summarised below according to [56].

The method makes use of reference models of the seven standard nucleobases (A, U, G, C, T, I, and P) generated from crystal structures stored in the Cambridge Structure Database [2]. A base-pair that is formed of two of those reference models represents an ideal, planar, and undistorted Watson-Crick base-pair. For each base a local right-handed coordinate system is generated.
2.4. ANNOTATING TEMPLATE CANDIDATES

(cf. Fig. 2.4). For a Watson-Crick base-pair the $x$-axes of both coordinate systems point in the direction of the major groove, along the perpendicular bisector of the vector $\vec{c}_1$ between the $C^*_1$-atoms of both bases. The origins are located at the intersection point of the $x$-axes with the vector between the pyrimidine-$C_6$ and purine-$C_8$ atoms. The $y$-axes point in the direction of the sugar-phosphate backbone, along $\vec{c}_1$.

Figure 2.4: Base-Pair Coordinate Systems for an $A$-$U$ pair. Cf. Sect. 2.4.1. Note that for a standard base-pair the origins of both coordinate systems overlap. For visualisation they have been separated in this figure.

Base-pair detection is then performed by fitting each of the standard bases to their corresponding base in the query structure. Modified residues are replaced by the best matching standard base. The occurrence of a base-pair is determined by computing the angle $\varphi$ and vertical distance $d_v$ between the planes in which the two bases are situated:

$$\varphi(i, j) = \frac{180^\circ \cdot \arccos(\vec{z}_i \cdot \vec{z}_j)}{\pi},$$

where $\vec{z}_i, \vec{z}_j$ are the normal vectors of the base-planes of bases $i$ and $j$ and $\cdot$ denotes the dot product.

$$d_v = ||\vec{n} \cdot \vec{d}||,$$

where $\vec{n}$ is the unit vector averaged over $\vec{z}_i, \vec{z}_j$, and $\vec{d}$ is the vector connecting the origins of both local coordinate systems.

Two $A$-$U$, $A$-$T$, $G$-$C$, or $I$-$C$ pairs of bases can form a canonical Watson-Crick base-pair, if all of the following criteria are fulfilled by the geometry of the given bases:

- $0^\circ \leq \varphi_x \leq 17^\circ$, where $\varphi_x$ denotes the angle between the two local $x$-axes.
- $157^\circ \leq \varphi_y \leq 180^\circ$, where $\varphi_y$ denotes the angle between the two local $y$-axes.
0° \leq \varphi_z \leq 30°$, where $\varphi_z$ denotes the angle between the two local $z$-axes and thus the angle between the two base-planes ($\varphi_z = \varphi$)

- $d_o < 2.5\text{Å}$, where $d_o$ denotes the distance of the two base-pair origins.

- $d_v < 1.5\text{Å}$, where $d_v$ denotes the vertical distance between the two base-planes.

The first three criteria are given by the definition of the standard Watson-Crick geometry.

If one or more of the above criteria are not fulfilled by the geometry of the examined bases, the bases can form a non-Watson-Crick base-pair if $\varphi \leq 65°$ and $d_v < 2.5\text{Å}$. Furthermore, there must exist at least one hydrogen bond between the two base ring-systems (donor-acceptor distance $< 3.4\text{Å}$). And at least one other hydrogen bond between either the base ring-systems ($(N,O)...(N,O)$ (donor-acceptor distance $< 3.75\text{Å}$) or $(N,O)...(C-H)$ (donor-acceptor distance $< 3.9\text{Å}$)), or between one base ring-system and the ribose-residue of the pairing base (donor-acceptor distance $< 3.75\text{Å}$). Pairs forming only two bifurcated hydrogen bonds between one donor and two acceptor atoms will only be assigned to one of the twelve standard non-canonical base-pairs if $\varphi < 50°$ and $d_v < 2.1\text{Å}$. Classification of the geometrical edges then occurs by counting the contact distances between the two bases.

Two bases that neither fulfill the criteria for a Watson-Crick nor for a non-Watson-Crick base-pair, might still be connected by a tertiary interaction. If there exists at least one hydrogen-bond between the ring-systems ($(N,O)...(N,O)$ (donor-acceptor distance $< 3.4\text{Å}$), $(N,O)...(O_2^*)$ (donor-acceptor distance $< 3.4\text{Å}$), or $(N,O)...(O_1P/O_2P')$ (donor-acceptor distance $< 3.2\text{Å}$)) of the given bases or between one base ring-system and the ribose-residue of the pairing base (donor-acceptor distance $< 3.1\text{Å}$).

Finally, the glycosidic bond configuration can be computed. Let $V_{12}$ be the vector pointing from the ring center of the first base to the ring center of the second base. Further, let $V_{1C}, V_{2C}$ be the vectors pointing from the $C_1^*\text{-atoms}$ of the bound ribose to their corresponding glycosidic-bound nitrogen atoms (pyrimidine: $N_1$, purine: $N_6$). Then the base-pair is in cis orientation, if and only if $(V_{12} \times V_{1C}) \cdot (V_{12} \times V_{2C}) \geq 0$. Otherwise, it is in trans orientation.

### 2.4.2 Stem Detection

During the stem detection step the base-pair information determined by the base-pair detection method is used to identify the stem-regions of a candidate molecule. The stem-detection algorithm (cf. Algorithm 1) detects the standard anti-parallel stems that form the major part of an RNA molecule’s secondary structure as well as parallel stems forming pseudoknot-motifs. The information resulting from the stem-detection process is essential for the reconstruction of the
2.4. ANNOTATING TEMPLATE CANDIDATES

complete secondary structure performed by the other methods of the annotation pipeline. In the following we describe the stem detection algorithm.

Given a secondary structure graph \( G = (V, E) \) the algorithm (cf. Algorithm 1) takes as input an ordered list of base-pairs

\[
\text{PAIRS} = E, \forall (u_i, v_i), (u_j, v_j) \in \text{PAIRS} : u_i < u_j \iff i < j.
\]

A stem \( \{([S = (u, v), T = (u', v')], \text{cross})\} \) consists of a starting \( S \) and terminating \( T \) edge and of a single flag-value \( \text{cross} \) stating whether the stem is crossing (i.e. violating the nested condition) the first detected stem (‘mother stem’) and thus forming a pseudoknot. The mother stem is not crossing itself, hence its cross-flag is not set.

As initial step, the algorithm uses the first edge \( (u_0, v_0) \in \text{PAIRS} \) to build a single-pair stem \( \{([u_0, v_0], (u_0, v_0), -1) \} \) (cf. Algorithm 1:1-3).

The algorithm then iteratively tries to fit each of the remaining base-pair edges in \( \text{PAIRS} \) to a stem in \( \text{STEMS} \). An edge \( (u, v) \in \text{PAIRS} \) is considered fitting to a stem \( \{([x, y], (x', y'), \text{cross})\} \in \text{STEMS} \) if either \( u - x' = 1 \text{ AND } v - y' = -1 \) (i.e. if both of its nodes \( u, v \) directly follow \( x' \) (or precede \( y' \) resp.) regarding the nucleotide backbone.) (cf. Algorithm 1:12-14 and Fig. 2.5(1)). Or if there is a bulge between either \( u \) and \( x' \) or \( v \) and \( y' \) (cf. Algorithm 1:15-17 and Fig. 2.5(2)). If no fitting stem can be found for the current edge then the algorithm builds a new single-pair stem from it (cf. Algorithm 1:18-20 and Fig. 2.5(3)).

Pseudoknot detection is performed in lines 9-11 (cf. Algorithm 1 and Fig. 2.5(4)). A base-pair edge \( (u, v) \) is considered a pseudoknot with a stem \( s = ([x, y], (x', y'), \text{cross}) \) if

\[
[x' < u \text{ and } u < y' \text{ and } y' < y] \text{ or } [x' > u \text{ and } u > y' \text{ and } y' > y]
\]

and if \( s \) does not enclose edges crossing the mother stem, i.e. does not form a pseudoknot itself.
Algorithm 1: StemDetection

Data: an ordered list of base-pair edges \( PAIRS \)

Result: a set of stem regions \( STEMS \)

1. \( STEMS \leftarrow \emptyset \)
2. \( \text{Edge} \ (u_0, v_0) \leftarrow \text{FRONT}[PAIRS] \)
3. \( STEMS \leftarrow \{((u_0, v_0), (u_0, v_0), \text{false})\} \)

4. for each edge \( (u, v) \in PAIRS \setminus \{(u_0, v_0)\} \) do
5. \( \text{is\_clustered} \leftarrow \text{false} \)
6. \( \text{is\_parallel} \leftarrow \text{false} \)
7. \( \text{last\_stem\_crossing} \leftarrow \text{false} \)
8. for each stem \( \left(\left[(x, y), (x', y')\right], \text{cross}\right) \in STEMS \) do
9. \( \text{if} \ [x' < u \text{ and } u < y' \text{ and } y' < y] \text{ or } [x' > u \text{ and } u > y' \text{ and } y' > y] \text{ then} \)
10. \( \text{is\_parallel} \leftarrow \text{true} \)
11. \( \text{if} \ \text{cross} = \text{false} \text{ then } \text{last\_stem\_crossing} \leftarrow \text{true} \)
12. \( \text{if} \ u - x' = 1 \text{ and } v - y' = -1 \text{ then} \)
13. \( \ (x', y') \leftarrow (u, v) \)
14. \( \text{is\_clustered} \leftarrow \text{true} \)
15. \( \text{else if } \text{BULGE}((u, v), (x', y')) \text{ then} \)
16. \( \ (x', y') \leftarrow (u, v) \)
17. \( \text{is\_clustered} \leftarrow \text{true} \)
18. \( \text{if not } \text{is\_clustered} \text{ then} \)
19. \( \text{if } \text{is\_parallel} \text{ and } \text{last\_stem\_crossing} \text{ then } c \leftarrow \text{true} \text{ else } c \leftarrow \text{false} \)
20. \( \text{STEMS} \leftarrow \text{STEMS} \cup \{((u, v), (u, v), c)\} \)
21. end

2.4.3 Secondary Structure Motif Assembly

The secondary structure motif assembly routine annotates secondary structure motifs using the previously determined stem data.

Definition: Secondary Structure Motif. A secondary structure motif is defined as a single stem (mother stem), the following unpaired residues (loop region), and all outgoing stems (child stems) of the loop region.

Figure 2.6 displays an example secondary structure with annotated stems. Given the above def-
2.4. ANNOTATING TEMPLATE CANDIDATES

inition, the structure consists of five secondary structure motifs: (S1, S2, S3, S4), S2, S3, (S4, S5), and S5.

Figure 2.6: An Example RNA Secondary Structure. S1-S5: The different stems as detected by the stem detection algorithm. (cf. Sect. 2.4 - Stem Detection)

The routine iteratively takes each stem (omitting single base-pair stems) and checks whether it and the following stem belong to a common secondary structure motif (cf. Sect. 1.1.1). If this is the case, the routine annotates a new structure motif to the corresponding stems. It then checks the next stems until it finds a stem that does not belong to the current motif. (cf. Fig. 2.7)

Figure 2.7: Sample Output of the Secondary Structure Motif Assembly.

Figure 2.7 shows the output of the secondary structure motif assembly for the example structure depicted in Figure 2.6. The routine starts with stem S1 which forms a multiloop with stems
CHAPTER 2. THE TEMPLATE FRAGMENT LIBRARY

S2, S3, and S4 (Step 1). It then detects the hairpin loops S2 and S3 as well as the internal loop consisting of S4 and S5 (Steps 2,3,4). Finally, the routine reaches hairpin S5 (Step 5) and terminates.

2.4.4 Region Identification

Starting from the results of the preceding secondary structure motif assembly the region identification step determines which unpaired residues belong to a given secondary structure motif. This step yields the nucleotide sequences for each motif. Each of the four different secondary structure motif types (hairpin loop, internal loop, multiloop, pseudoknot structure) consists of a certain number of connected residue chains (hairpin: 1, internal: 2, n-multiloop: n+1, pseudoknot: 1).

Figure 2.8: Subsequence Regions for Different Secondary Structure Motif Types. From left to right: 3-multiloop, hairpin-loop, internal loop, pseudoknot.

Figure 2.8 displays the division of the four secondary structure motif types into subsequence regions. The 3-multiloop-structure is divided into four regions ((a, b), (b', c), (c', d), (d', a')), the internal loop into two ((a, b), (b', a')), and the hairpin and pseudoknot structures into one ((a, a') or (a, b'), resp.).

2.4.5 Tertiary Structure Motif Assembly

This is the last step of the motif annotation process. The remaining (tertiary contact) base-pairs which have not been processed during the stem detection step are used to build up tertiary structure motifs. Algorithm 2 describes how the tertiary structure motifs are built by the assembly routine.

Building tertiary structure motifs from the given information is simple. Each remaining base-pair is tested against the set of secondary structure motifs resulting from the preceding steps. If the base-pair is an intra-motif contact, i.e. both of its bases belong to the same motif, then it does not induce the formation of a new tertiary structure motif as the three-dimensional structure information is already part of the existing secondary structure motif. Thus, the base-pair can be disregarded by the assembly step without loss of information (2:3-4).
If the base-pair is formed between residues of two different secondary structure motifs, it is used to build a new tertiary structure motif (2:7-8) if and only if the two secondary structure motifs have not already formed an tertiary structure motif discovered earlier (2:5-6), i.e. if multiple base-pairs connect the same pair of secondary structure motifs.

**Algorithm 2**: Tertiary Structure Motif Assembly  

**Data**: a list of secondary structure motifs \( \text{MOTIFS} \), a list of tertiary contacts \( \text{CONTACTS} \)  

**Result**: a set of tertiary structure motifs

1. **for each contact** \( c \) **in** \( \text{CONTACTS} \) **do**
2.   **Identify** the secondary structure motifs \( r_1, r_2 \) **connected** by \( c \)
3.   **if** \( r_1 = r_2 \) **then**
4.     **Ignore** the contact.
5.   **else if** \( \exists \) a motif \( (r_1, r_2) \) **in** \( \text{MOTIFS} \) **then**
6.     **Ignore** the contact.
7.   **else**
8.     **Add** a new motif \( (r_1, r_2) \) **to** \( \text{MOTIFS} \).
Chapter 3

RNA Threading Methods and Algorithms

This chapter describes the methods and algorithms employed in the threading framework. The Lagrangian-Relaxed RNA Structure Alignment (Lara) algorithm (Sect. 3.2) is a sequence-structure alignment method. It serves as the core (threader routine) of the threading framework, aligning template fragment structures to target sequences. The hairpin detection (Sect. 3.3) method is an auxiliary routine to optimise the search for short sequences, e.g. small fragment structures such as hairpins. The alignments resulting from the Lara algorithm are postprocessed by the contact validation via isostericity matrices (Sect. 3.4), validating the aligned base-pairs of template and target. The chapter closes with a description of a fragment assembly approach based on clique detection (Sect. 3.5).

3.1 The Threading Framework

The threading framework (cf. Fig. 3.1) consists of a number of C++ programs controlled by a manager script written in Python. The heart of the framework – the threader routine – is an RNA sequence-structure alignment (Lara) method which has been described in Sect. 3.2. Searches performed by the framework can either target at finding large structures such as internal or multi-loops or even complete secondary structures. Or the search can be aimed at finding small structures, i.e. hairpin loops (hairpin search). Depending on the type of search, a work-library consisting of the respective templates has to be composed.

The framework takes as input an RNA sequence (target sequence), the selected work-library, and optionally a predicted or annotated secondary structure (target structure). If no target structure is provided, the target sequence is submitted to a secondary structure prediction/annotation routine (currently RNAfold [22]) to generate base-pair probability matrices.

25
The structure alignment algorithm is not capable of aligning short structures to long sequences without scattering the shorter structure along the sequence. Therefore, the search for hairpins cannot be performed on the complete target sequence. Instead, if a hairpin search is to be performed, the target sequence is submitted to the hairpin detection subroutine (cf. Sect.3.3). This subroutine scans the target sequence for candidate subsequences that could possibly fold into hairpin structures, thus minimising the chances of misaligning a template to a region not capable of folding into a hairpin. The target sequence is subsequently aligned against each template structure. In case of hairpin searches each found candidate subsequence of the target is aligned against each template structure. The resulting sequence/structure mappings (alignments) of template and target are then examined and scored by the contact validation routine. This routine ensures that the target sequence is capable of forming any contact contained in the aligned template structure (cf. Subsect. 3.4). The validated sequence/structure mappings finally serve as input for
optional postprocessing steps, e.g. fragment assembly (Sect. 3.5) or 3D structure assembly via molecular dynamics simulation.

The framework has been designed with respect to the exchangability of its individual components. Thus, any subroutine of the framework (yellow boxes) can be easily upgraded or even exchanged with a more efficient method.

3.2 Lagrangian-Relaxed RNA Structure Alignment (Lara)

The current framework implementation uses Lara as the threader routine. In the following, we describe the Lara algorithm and its prerequisites.

**Definition:** Let $S$ be a sequence $s_1, \ldots, s_n$, $s_i \in \Sigma = \{A, C, G, U\}$.

(i) An ordered pair $(i, j)$ is called an *interaction* if $s_i, s_j \neq -$, $(s_i, s_j)$ forms a Watson-Crick base-pair, and $j - i > 3$, i.e. the paired bases are not too close to each other.

(ii) A set $P$ of interactions $(i, j)$ of $S$ is called an *annotation* of $S$. An annotation $P$ that contains all possible interactions of $S$, is called a *complete annotation*.

(iii) Two interactions $(i, j), (k, l)$ are said to be *conflicting* if they touch each other (i.e. they share a base), and non-conflicting otherwise.

(iv) If all interactions $(i, j) \in P$ are non-conflicting, then $P$ is called a *secondary structure* of $S$.

(v) A secondary structure $P$ is called *nested* if all interactions $(i, j) \in P$ are non-crossing. I.e., if it holds for all $(i, j), (k, l) \in P$ that either $i < j < k < l$, i.e. $(i, j)$ precedes $(k, l)$ or $i < k < l < j$, i.e. $(i, j)$ includes $(k, l)$ (cf. Fig. 3.2). If two interactions $i_1, i_2 \in P$ induce a non-nested secondary structure, either $i_1$ or $i_2$ is formed by a tertiary interaction (cf. section 1.1.2).

(vi) The pair $(S, P)$ is called an *annotated sequence*.

![Diagram](image)

Figure 3.2: RNA Secondary Structure: The interactions $i = (1, 9), j = (2, 5), k = (6, 8), l = (4, 7)$ are nested, with $j$ preceding $k$ and $i$ including $j, k$, and $l$. The interactions $j, k$, and $l$, however, are not nested, as the preconditions for a nested secondary structure hold for neither pair of them. Please note that this is not a valid secondary structure at all, as the interactions $j, k$, and $l$ are too close to each other.
The following definition for an RNA structure alignment (RSA) based on Lagrangian relaxation is given according to [5, 6].

**Defining RSA as a Graph-Theoretic Problem**

Given two annotated sequences \((S_1, P_1), (S_2, P_2)\) as input a graph \(G = (V, A \cup I)\) can be defined as follows: \(V = S_1 \cup S_2\) denotes the set of the vertices of \(G\). \(A\) is a set of alignment edges which are edges that connect vertices \(s_{1,i} \in S_1\) and \(s_{2,j} \in S_2\) and \(I\) denotes a set of interaction edges between vertices of the same sequence (Figure 3.3). Two alignment edges are said to be conflicting if they cross or touch (i.e. they share an incident vertex) each other. A subset \(A \subset A\) which contains only alignment edges that are not conflicting, is called an alignment.

An ordered pair of interaction edges \((i, j)\) with \(i = (i_1, i_2) \in P_1\) and \(j = (j_1, j_2) \in P_2\) is called an interaction match. An interaction match \((i, j)\) is said to be realised by an alignment \(A\) if and only if the alignment edges \((i_1, j_1)\) and \((i_2, j_2)\) are contained in \(A\). (Figure 3.3)

Realisation of an alignment edge or interaction match yields a benefit. Therefore positive weights are assigned to each alignment edge and to each interaction match. An alignment edge \(m\) is weighted by a weight \(w_m\) corresponding to e.g. the similarity score of the bases connected by \(m\). An interaction match \((l, m)\) is assigned a weight \(w_{lm}\) corresponding to e.g. the base pair probability for the bases forming \((l, m)\) computed by e.g. RNAfold [22].

**Formulation of an Integer Linear Program (ILP) for RSA**

An alignment of two RNA-sequences taking into account not only sequential but also structural information, is an optimization problem trying to maximise the score that results from realisation.
of alignment edges and interaction matches. To form a valid RNA secondary structure alignment the following conditions have to be satisfied [6]:

1. Each vertex is incident to at most one interaction edge, i.e. a base can only form an interaction with at most one other base.

2. Any two alignment edges are non-conflicting.

3. An interaction match \((l, m)\) is realised by an alignment, if and only if the interaction edges \(l, m\) are connected by alignment edges. I.e., \((l, m)\) will only be realised, if the bases forming \((l, m)\) are aligned.

Let \(l, m\) be alignment edges, let \(I\) denote the set that contains all subsets of conflicting interaction edges \(l \in I\), and let \(x_m, y_{lm}\) be variables with

\[
x_m = \begin{cases} 
1 & \text{if the alignment edge } m \text{ is realised by } A \\
0 & \text{else} 
\end{cases}
\]

(i.e. only the weights of those alignment edges realised by \(A\) contribute to the final sequential score) and

\[
y_{lm} = \begin{cases} 
1 & \text{if } l, m \text{ realise the interaction match } (l, m) \\
0 & \text{else} 
\end{cases}
\]

(i.e. two alignment edges contribute their weights to the final structural score if and only if they realise an interaction match).

This yields the following ILP-formulation [5, 6]:

\[
\max \sum_{m \in A} \sum_{l \in A} w_{lm} y_{lm} + \sum_{m \in A} w_m x_m \tag{3.1}
\]

such that:

\[
\sum_{l \in I} x_l \leq 1 \quad \forall I \in I \tag{3.2}
\]

\[
y_{lm} = y_{ml} \quad \forall l, m \in A, l < m \tag{3.3}
\]

\[
\sum_{l \in A} y_{lm} \leq x_m \quad \forall m \in A \tag{3.4}
\]

\[
x, y \in [0, 1] \tag{3.5}
\]

Constraint (3.2) ensures that realised alignment edges are non-crossing (for each subset \(I \subset \mathcal{I}\) at most one \(l \in I\) exists, such that \(x_l = 1\). (3.3) and (3.4) assert that each vertex is incident to at
most one interaction edge. In constraint (3.3) the order \( l < m \) denotes an arbitrary order on the elements of \( A \). It makes sure that identical constraints show up only once in the ILP. The ILP stated above is \( NP\text{-}hard \), and therefore we cannot hope to solve it in polynomial time. Relaxing the ILP by simply dropping constraint (3.3), however, results in qualitatively rather poor solutions for the original ILP. Therefore Bauer and Klau propose to solve the ILP via Lagrangian Relaxation [5, 6].

Solving the Problem via Lagrangian Relaxation

The Lagrangian Relaxation method can be used to solve constrained problems where the constraints can be divided into two sets: the ”good” constraints, which do not complicate the solution of the problem, and the ”bad” constraints, which make it very hard to solve. The Lagrangian relaxed problem results from relaxing the problem by the ”bad” constraints by moving them into the objective function and assign penalty terms to them [5].

Constraint 3.3 of the ILP is such a ”bad” constraint: instead of dropping it, we move it into the objective function and assign a penalty term to the constraint: violating the constraint yields a reduction of the value of the objective function. This leads to the following formulation of the Lagrangian relaxed problem:

\[
\text{max} \sum_{m \in A} \sum_{l \in A} w_{lm} y_{lm} + \sum_{m \in A} w_m x_m + \sum_{l \in A} \sum_{m \in A, l < m} \lambda_{lm} (y_{lm} - y_{ml})
\]  

(3.6)

with constraints (3.2), (3.4) and (3.5).

This has the effect that a penalty given by \( \lambda_{lm} \) is applied to the overall score, if \( y_{lm} = y_{ml} \) does not hold for all \( l, m \in A, l < m \). Consequently, a \( \lambda_{opt} \) providing an optimal bound to the original problem has to be found. This can be achieved by initially setting \( \lambda_{ml} = -\lambda_{lm} \) for \( l < m \) and \( \lambda_l = 0 \) which yields

\[
\text{max} \sum_{m \in A} \sum_{l \in A} (\lambda_{lm} + w_{lm}) y_{lm} + \sum_{m \in A} w_m x_m
\]  

(3.7)

The variable \( \lambda_{opt} \) can now be computed via iterative subgradient optimisation, which is a common method in non-differentiable optimisation [21].

Lara: The algorithm

The algorithm takes as input two RNA sequences and their corresponding secondary structure information. The secondary structure information provides the weights for the interaction matches. It can be obtained from different methods, such as secondary structure prediction (e.g. RNAfold
3.2. LAGRANGIAN-RELAXED RNA STRUCTURE ALIGNMENT (LARA)

[22]), computation of the base-pair probability matrix [37], or manual annotation. The base-pair probability matrix for a sequence $S$ is an annotation containing all possible interactions in $S$ with each interaction weighted by the probability of the interaction to be formed. The advantage of using a complete annotation – as presented by a base-pair probability matrix – over a predicted secondary structure, is explained by the fact that a secondary structure prediction might not include all base-pairs. Also a base-pair probability matrix includes probabilities for tertiary contacts between bases, theoretically allowing the integration of tertiary structure into the alignment. Manual annotation or tweaking of computed predictions or annotations can be used to force certain base-pairs to be incorporated into the structure which is realised by an alignment.

The initial phase of the algorithm gathers the data needed for computing the alignment:

The first step is to collect a set of alignment edges. We compute a Needleman-Wunsch global sequence alignment [40] and insert all alignment edges that score within a given range below the optimal solution [45]. Next, the interaction edges are computed using the structure annotations. The weights for each edge are then determined. Alignment edges are weighted according to their traditional sequence score. Interaction matches are weighted according to their base pair probabilities which have been supplied as input to the algorithm.

An upper bound $U$ to the problem is computed in a two-stage procedure. First, we compute the maximum profit that each alignment edge can achieve. We take these scores as the input for a traditional sequence alignment algorithm, yielding an upper bound on the original problem. The actual score is given by a lower bound $L$ that is computed by a matching of maximum weight on the alignment [5].

A structural alignment is then computed iteratively. If upper and lower bound merge into approximately the same value (allowing a small numerical deviation $\delta$) after some iteration, the algorithm terminates, as the optimal solution has been found then. Else the algorithm runs until the maximum number of iterations has been reached. The overall running time of $O(|A|^2)$ is dominated by the global sequence alignment.

A simple scheme of the iteration phase of the Lara algorithm:

1. Computation of the maximal profit of each alignment edge and alignment (resulting in an upper bound $U$)
2. Computation of an alignment weighted with the previously computed profits
3. Computation of a feasible solution given the alignment edges from the preceding step
4. Computation of subgradients and adaptation of the Lagrangian multiplier
   If $|U - L| <= \delta$ then terminate.
   Goto 1.
Employment of Lara as Threader Routine

The Lara algorithm computes an optimal global structure alignment for two continuous RNA structures. This poses a problem, as most of the structure fragment types contained in the template library are not continuous but rather consist of two (internal loops, bi-loops, and certain tertiary bi-motifs) or more (multiloops, tertiary bi-motifs) non-connected subsequence regions: following the definition in Sect. 2.4.3, a fragment includes one stem, the following unpaired loop region, and \( n \) outgoing (child) stems, yielding \( n + 1 \) subsequence regions. Between the end of the \( n \)th and the start of the \( (n + 1) \)th sequence (cut-site) there have to be at least three bases (the minimum size for a hairpin loop) which are not part of the fragment itself. The discontinuous fragments cannot be processed by the Lara algorithm. Hence, the subsequence regions forming such a fragment have to be concatenated to a single template structure preserving the cut-site information.

The solution to this problem is to introduce an additional placeholder character, say \( \bar{X} \), that is not included in the standard nucleotide alphabet. Ideally, \( \bar{X} \) would serve as a wildcard character matching any number of bases without a penalty. For simplicity, we insert an \( n_X \)-length sequence of \( \bar{X} \)'s into the template at each cut-site instead. In addition, we modify the scoring matrices to reward a match of a nucleobase-character against \( \bar{X} \).

3.3 Hairpin Detection

One of the drawbacks of employing Lara as alignment algorithm is its incapability of finding small motifs, i.e. hairpins in a relatively long target sequence. Previous tests dealing with the alignment of hairpin-length structures to an RNA sequence have shown that the hairpin structure will either be scattered along the longer sequence [49] (cf. Fig. 3.4) or it will be aligned as a whole to the 5´-end of the target (cf. Section 4.2.1). As hairpins are an essential part of each RNA structure, another solution has to be found to cover the hairpin alignment. One approach is to find regions in the target sequence which possibly could fold into a hairpin-loop and separately align these regions to the template library.

May et al. have developed an approach to detect miRNA signals (i.e. stem regions with certain properties) within genomic sequences without having to actually fold the sequence, thus saving computation time [36]. Their method can be used to locate hairpin-candidate regions by searching for subsequences of the target sequence that fulfill the following criteria: The subsequence has to (1) contain a number of nucleotides similar (within a given \( \delta \)) to the length of the template hairpin. (2) It has to contain a cross-palindromic part of a length similar to the stem length of the template hairpin, i.e. the ends of the sequence must be able to form a stem consisting solely of canonical and GU-wobble base-pairs. In addition to these two criteria several filters are employed.
3.3. HAIRPIN DETECTION

Figure 3.4: Alignment Scattering. A hairpin structure (bottom, light green) is aligned against a tRNA structure (top). The hairpin is identical to the anticodon-hairpin of the tRNA (highlighted in dark green). Hence, it would be expected that both hairpins are aligned with each other. However, in the depicted alignment the hairpin is misaligned to the tRNA 5'-end and scattered along the whole length of the tRNA structure.

The approach uses a modified Smith-Waterman [51] dynamic programming local alignment algorithm to detect and score stem regions in an input sequence. Let $hp$ be a hairpin-loop template of length $|hp|$, $s$ the length of the stem region of $hp$, and $t$ be the target sequence. The algorithm slides a window of size $w = |hp|$ along $t$. At position $i < |t| - w$ the subsequence $t[i,i+w]$ is then tested for being a possible candidate region: The reversed sequence $t[i+s,i]$ is locally aligned against $t[i+s,i+w]$ (cf. Fig. 3.5) using a modified scoring scheme that regards canonical (A-U, G-C) and wobble pairs (G-U) as match instead of checking for sequence similarity.

Figure 3.5: Hairpin Detection by Local Alignment. The window $w$ is slided along the sequence $t$. The subsequence of $t$ contained within $w$ is reversely aligned against itself. Pipe symbols depict detected base-pairs forming a stem. Asterisks depict base-pair candidates that have been falsely detected.

After overlapping regions have been removed only the best scoring regions are kept as candidates. These are folded with RNAfold to discard regions that do not adapt a hairpin structure or display a predicted minimum free energy above a certain threshold. The remaining candidates can then be further processed.
CHAPTER 3. RNA THREADING METHODS AND ALGORITHMS

3.4 Contact Validation (Isostericity Matrices)

As described in Sect. 1.1.1, base-pairs (contacts) can be exchanged against each other (e.g. the cis-Watson-Crick C-G and A-U pairs) without disturbing the native three-dimensional structure of an RNA molecule if they are isosteric. In contrast, the scoring scheme used by the employed alignment algorithm does not differentiate between different base-pair types, possibly aligning non-isosteric base-pairs (e.g. the cis G-U and U-G wobble pairs). Using such an alignment during the assembly of a structure model for a given target sequence forces the target sequence to adopt the base-pair structure of the template, thus leading to a three-dimensional model structure which could not naturally be formed by the bases of the target sequence. If the base-pairs are isosteric it can be assumed that the target sequence can actually form the base-pair.

The validation of aligned secondary structure base-pairs could be implemented by penalising or even prohibiting the alignment of non-isosteric base-pairs within the scoring scheme of the employed alignment algorithm. The method is not only limited to secondary structure base-pairs. It can also be used to validate the alignment of tertiary structure contacts possibly included in the template structure. However, due to the nature of the employed alignment and the limited availability of tertiary structure prediction programs, it is currently not possible to integrate the validation directly into the scoring scheme. Instead, the resulting alignments are postprocessed by the contact validation routine: for each contact contained in a template structure we compare the respective template bases and their aligned target bases against an isostericity matrix. If the target bases can form a base-pair that is isosteric to the contact base-pair, then we assume that the target sequence can possibly include the contact. As an example, consider a trans U-G base-pair that is formed between the Watson-Crick edges. The pair has been aligned to an A and a C in the target sequence. According to the respective isostericity matrix these types of base-pairs are isosteric to each other. Thus, the contact is valid and as such included in the target structure model.
3.5 Fragment Assembly via Clique Detection

The alignments resulting from a threading run provide mappings of different templates onto the target sequence. These mappings show which parts of the target sequence have been covered by the threading. However, the mappings of two different templates are completely unrelated to each other: while the threading might have been able to find template fragments to cover the complete structure of the target sequence, the distinct fragments still have to be assembled to form a final structure model. This section describes an graph-theoretic approach to the assembly problem.

Given a target sequence and a set of template structure fragments which have been aligned to the target sequence. Each aligned fragment covers a part of the target sequence, possibly overlapping with other fragments. E.g., consider a hairpin fragment and an internal loop fragment: if the hairpin and the outgoing stem of the internal loop have been aligned to the same part of the target sequence, then the two fragments overlap. An aligned fragment is defined by its anchor points (cf. Fig. 3.6). These anchor points correspond to the indices of certain bases (target bases) within the target structure $T$ to which the fragment $F$ has been aligned. Each aligned fragment has at least two anchor points (Fig. 3.6a.): $a_{\text{start}}$, the index of the base of the target sequence which has been aligned to the first base of the fragment, and $a_{\text{end}}$, the index of the target base which has been aligned to the last base of the fragment. Fragments consisting of multiple regions (chains) such as multiloops have additional inner anchor points (Fig. 3.6b.). These inner anchor points are given as pairs $(a_i, a_{i+1})$, corresponding to the two target bases $a_i$ and $a_{i+1}$ which have been aligned to the bases on the boundaries of two subsequent chains. Following the motif definition given in Sect. 2.4.3, all anchor points ($a_{\text{start}}$, $a_{\text{end}}$, and all inner anchor points) have descriptive properties: $a_{\text{start}}$ and $a_{\text{end}}$ correspond to the first base-pair of the mother stem of the fragment and the pairs of inner anchor points describe the respective last base-pair (cut-site) of its child stems.

Two distinct fragments $F_1, F_2$ with anchor points $a_{\text{start}}^1, a_{\text{end}}^1$ and $a_{\text{start}}^2, a_{\text{end}}^2$ overlap, if (without loss of generality)

$$a_{\text{start}}^1 \leq a_{\text{start}}^2 \leq a_{\text{end}}^1 \leq a_{\text{end}}^2$$

(i.e., $F_2$ starts "inside" of $F_1$ and ends "outside" of $F_1$) or if $F_2$ is contained in $F_1$:

$$a_{\text{start}}^1 \leq a_{\text{start}}^2 \leq a_{\text{end}}^2 \leq a_{\text{end}}^1.$$
(Fig. 3.7b.), i.e.

\[ \exists (a_i^1, a_{i+1}^1) \in I : \quad a_i^1 \leq a_{\text{start}}^2 \leq a_{\text{end}}^2 \leq a_{i+1}^1 \uplus a_{\text{start}}^1 \leq a_i^1 \leq a_{i+1}^1 \leq a_{\text{end}}^2 \]
\[ \land \forall (a_j^1, a_{j+1}^1) \in I, i \neq j : \quad \neg(a_j^1 \leq a_{\text{start}}^2 \leq a_{\text{end}}^2 \leq a_{j+1}^1 \uplus a_{\text{start}}^1 \leq a_j^1 \leq a_{j+1}^1 \leq a_{\text{end}}^2) \]

Figure 3.6: Fragment Assembly: Anchor Points and Overlaps. cf. Sect.3.5 a.) Anchor points of fragment \( F \) with respect to target sequence \( T \). b.) Inner anchor points of fragment \( F \) with respect to target sequence \( T \). c.) Examples of overlapping fragments.

The general aim of an assembly approach is to find a set of aligned template structure fragments, such that a large fraction of the target sequence is covered. I.e., we attempt to create a structure model covering as many parts of the target sequence as possible, using only compatible fragments. This is justified by the fact that the structure model for the target is built utilising the secondary
structure as a scaffold or guide. Using incompatible fragments leads to nonsensical mapping of paired regions to unpaired ones and vice versa.

![Figure 3.7: Fragment Overlap Types.](image)

Overlapping or respectively non-overlapping regions, are highlighted by dashed boxes.

The assembly problem can be solved by creating a compatibility graph \( G = (V, E) \), with vertices \( v \in V \) corresponding to the fragments in the set of template fragments \( \mathcal{F}^T \) aligned to the target sequence \( T \) (Algorithm 3:3-4). We add an edge \( (u, v, \text{type}) \) between two vertices \( u, v \) to \( E \), if and only if either \( F^T_u, F^T_v \in \mathcal{F}^T \) do not overlap (i.e., a structure can contain multiple substructures which do not overlap (e.g. hairpins), Algorithm 3:7, \( \text{type} = 0 \)) or if \( F^T_u \) and \( F^T_v \) form a compatible overlap (Algorithm 3:11, \( \text{type} = 1 \)) as defined above. \( G \) then contains all possible (compatible) combinations of the distinct fragments in \( \mathcal{F}^T \). Note that \( F^T_u \) and \( F^T_v \) do not form a compatible overlap if they start and end at the same anchor points (Algorithm 3:9). Accepting these fragments as being compatible would imply mapping two different fragments on the same target region. The subgraphs of \( G \) correspond to different combinations of fragments. All complete subgraphs (cliques) of \( G \) represent partial or complete structure models for \( T \). Structure models that consist of the maximum number of fragments correspond to the cliques of maximal size. The larger the size of a clique the more regions within the target are covered by the corresponding template fragments. E.g., for a tRNA target a clique of size 3 (one hairpin, one bi-loop, one multiloop) or 4 (three hairpins, one multiloop) – depending on the employed template library – can be sufficient to cover the complete structure of the target.

Unfortunately, finding the maximum clique(s) in a graph is NP-hard [24]. However, we can obtain all cliques contained in \( G \) by running the branch-and-bound Bron-Kerbosch clique-detection algorithm [13]) on \( G \). This algorithm finds all cliques within an undirected graph. The resulting set of cliques as yielded by the Bron-Kerbosch algorithm corresponds to a set of unscored mappings. Note that determining the quality of a structure model, i.e., finding the best global alignment is – as in protein threading – not yet solved. These mappings can be used as a scaffold or guide for the assembly of three-dimensional structure models for the target sequence. To cover as many parts of the target as possible, we only take cliques of maximum size into account when creating a structure model. Figure 3.8 schematically displays the process of modeling aligned template fragments as a compatibility graph.
Algorithm 3: FragmentAssembly

**Data:** a set $\mathcal{F}^T$ of template fragments aligned to a target sequence $T$  
**Result:** a set $\mathcal{C}^T$ containing the maximal cliques of $\mathcal{F}^T$

1. $\mathcal{C}^T \leftarrow \emptyset$
2. Graph $G = (V, E)$
3. for each fragment $F = (a_{\text{start}}^F, a_{\text{end}}^F, (a_1^F, a_2^F), \ldots, (a_{n-1}^F, a_n^F)) \in \mathcal{C}^T$ do
   1. $V \leftarrow V \cup \{F\}$
4. for each vertex $v \in V$ do
   5. for each vertex $u \in V \setminus \{v\}$ do
      6. if $a_{\text{end}}^u < a_{\text{start}}^v$ then $E \leftarrow E \cup \{(u, v, 0)\}$
      7. else if $a_{\text{start}}^u \leq a_{\text{start}}^v$ and $a_{\text{end}}^u \leq a_{\text{end}}^v$ then
         8. if $a_{\text{start}}^u = a_{\text{start}}^v$ and $a_{\text{end}}^u = a_{\text{end}}^v$ then continue
      9. for each pair $p = (a_i^v, a_{i+1}^v)$ of inner anchor points of $u$ do
         10. if $a_i^v \leq a_{\text{start}}^v \leq a_{\text{end}}^v$ or $a_{\text{start}}^v \leq a_i^v \leq a_{\text{end}}^v$ then
             11. $E \leftarrow E \cup \{(u, v, 1)\}$
             12. break
13. $\mathcal{C}^T \leftarrow \text{BRON-KERBOSCH}(G)$
Figure 3.8: Graph-Based Fragment Assembly. Top: Secondary structure bracket notation for a sample target and the four structure motifs that are contained within the target. Asterisks represent unpaired regions that are not covered by a fragment. Center and Bottom: A graph is created, based on the overlap-relations between the different template fragments used for the modeling of the target. All complete subgraphs (cliques) of the graph contain a structure model for the target sequence. Coloured vertices represent template fragments used to model target motifs which are highlighted with a similar colour. Bold edges depict a clique of size 4, corresponding to a possible structure model for the target, blue edges connect overlapping fragments, pink edges connect non-overlapping fragments. Grey vertices represent other theoretical template fragments, which have not been specified more closely.
Chapter 4

Computational Results

4.1 The Template Fragment Library

The template fragment library provides data for fragment-based RNA modeling approaches. It consists of fragments corresponding to RNA secondary structure motifs and tertiary structure bi-motifs, e.g. *kissing hairpins*. Fragments are stored as contact maps, including primary, secondary, and tertiary structure information (cf. Fig. 2.2). This section describes the results of the database creation and provides some statistics of the library.

4.1.1 Library Creation

As mentioned in Section 2.1, the data foundation for the template fragment library is acquired from the Brookhaven Protein Data Bank (PDB). The PDB is queried for all entries containing at least one RNA chain, yielding a total of 987 PDB entries. This initial data set does of course contain a lot of data with only very short RNA sequences and large amounts of non-ribonucleic acid molecules.

The PDB data is examined for its content and filters are applied on the complete data set. Filtering and reviewing is performed semi-automatic. Manual reviewing of the filter results ensures that only valid RNA structures are introduced into the fragment library. 65 database entries are filtered out because they contain DNA which prohibits the processing of the respective entry. E.g. two DNA and RNA chains can form a hybrid helix by inter-chain cis-Watson-Crick (cWC) pairing. Such hybrid helices cannot be classified correctly by the downstream secondary structure processing routines (a helix combines secondary and tertiary structure as it contains of a set of cWC base-pairs (a typical secondary structure *stem-motif*) formed between two distinct chains (quarternary structure).
CHAPTER 4. COMPUTATIONAL RESULTS

A further set of 20 entries is removed because they contain unidentifiable residues within an RNA chain. A large set of data is removed after the secondary structure detection step. For 311 entries the detection routine cannot compute a valid secondary structure. A typical reason for this is backbone-only-data, i.e. the data does not contain the atom coordinates for the base ring systems which are needed for secondary structure detection. A set of 13 more entries can not be processed by the fragment building routines for various reasons. After filtering and processing, the data foundation contains 578 source structures.

Figure 4.1: Distribution of RNA Species - Source Molecules

4.1.2 Library Statistics

This subsection provides an overview over the data stored in the fragment template library. The fragment source data (i.e. the individual PDB entries) originate from a wide range of different RNA species (cf. Figure 4.1). As described by Boit in 2003 [10] most of the entries (68%) originate from one of the three major RNA species: tRNA, rRNA, and viral RNA (vRNA). In contrast to that prior observation, the tRNA-species (17% - 103 entries) no longer supplies the largest fraction of the public available RNA data. It is displaced by the rRNA- and vRNA-species, supplying 27% (156 entries) or respectively 24% (146 entries) of the total number of entries. The fourth largest fraction (9% - 55 entries) is unclassified data. This fraction represents PDB entries which do not contain a definite classification of their species. The remaining data is collected from the
4.1. *THE TEMPLATE FRAGMENT LIBRARY*

ribozyme (7% - 42 entries), mRNA (21), aptamer (20), RNAse P-related RNA (13), srpRNA (11), and intron (11) species. The remaining 20 entries include members of the snRNA (6), telomerase RNA (5), riboswitches (4), transfer-messenger RNA (2), double-stranded (ds) RNA (1), snoRNA (1), and exon(1) species.

The fragment template library contains 24977 secondary structure and tertiary bi-motifs. (cf. Fig. 4.3 and Fig. 4.2). A part of the structures that could not be processed completely due to erroneous source data, misclassification due to pseudoknotted areas, or other difficulties, is contained within the library. A major part (90%) of the structures originates from rRNA source molecules. tRNA and vRNA source molecules only contribute 3% resp. 2% of all fragments. The remaining 5% are distributed between the other species.

The secondary structure fragments can be classified after the commonly known RNA secondary structure loop classes (s. Sect. 1.1.2 and Figure 1.6). Tertiary structure motifs are divided into bi-loop motifs (i.e. interactions between the unpaired regions of two distinct hairpins (*kissing hairpins*)) and other bi-motifs (formed by tertiary contacts between two secondary structure motifs, e.g. stem-loop interactions). Note that tertiary structure motifs consist of secondary structure motifs which are contained in the fragment library as single fragments. However, these are not redundant entries, as a secondary structure motif belonging to a tertiary structure motif can be used to model another single secondary structure motif as well. The majority of the entries belongs to the hairpin (28% - 6988 fragments) and internal loop classes (32% - 8028 fragments). Tertiary bi-motif fragments constitute the third largest fraction (21% - 5272 fragments). The fourth largest fraction (13% - 3273 fragments) are those fragments which have been classified as multiloops. Finally, pseudoknot-like structures and bi-loop motifs represent 6% of all fragments (pseudoknots 4% - 984 fragments, bi-loops: 2% - 432 fragments).
CHAPTER 4. COMPUTATIONAL RESULTS

Figure 4.2: Distribution of RNA Species - Fragments

Figure 4.3: Distribution of Fragment Types
4.2 Testing of Framework Components

This section describes the performed tests of two framework components: the applicability of the Lara alignment algorithm as threader routine (Subsect. 4.2.1) and the self-recognition capability of the contact validation via isostericity matrices (Subsect. 4.2.2).

4.2.1 Lara Alignment Validation

To determine whether the Lara algorithm is capable of serving as threader routine within the threading framework, i.e. if it is able to find matching regions between target and template, we perform test alignments. First, we attempt to find a multiloop fragment in its source structure. If the alignment algorithm fails finding the multiloop by aligning it to the corresponding (identical!) regions in the source structure, it can obviously not be employed as a threader routine. The second test aims at finding a multiloop motif in a different tRNA target structure. If the algorithm is capable of aligning the multiloop to feasible regions within the target structure (i.e. it should at least align the mother stem correctly to the corresponding regions at the 5’- and 3’-ends of the target structure, as the hairpin regions can be modeled separately), it can in general be employed for the RNA threading problem. The final test attempts finding a hairpin in its source structure. Aligning short structures – such as hairpins – to a long target sequence might pose the problem of alignment scattering. Hence, this alignment type has to be tested separately. This test is performed to show that the Lara algorithm cannot correctly compute this type of alignment. It explains the necessity of an additional method to perform hairpin alignments: the hairpin detection described in Sect. 3.3.

Finding a multiloop fragment in its source structure

This first test attempts at finding the 3-multiloop (2tra:0-11;22-30;38-51;59-70) of a transfer-RNA in its full source structure. A secondary aim of this search is to determine a feasible value for the length $\bar{n}_X$ for the cut-site sequence (cf. Sect. 3.2) such that the alignment can align the multiloop fragment to the corresponding regions in the source structure. Figure 4.4 displays the results for $n_X = 0, 1, 3, 5, 10$.

The first four alignments share a striking similarity: The leftmost child stem is not (or for $n_X = 5$ only poorly) realised by either of the alignments. The explanation for this lies within the secondary structure prediction for the target. The leftmost child stem has not been predicted by the prediction routine. (cf. Fig. 4.5)

In contrast, the mother stem and rightmost child stem are almost fully covered by the corresponding secondary structure motifs within the target. For $n_X = 10$ the secondary structure of the fragment is completely realised by the alignment. This alignment is the only one with gaps
**Figure 4.4:** Test Alignments for 2tra. Alignments for \( n_x = 0, 1, 3, 5, 10 \). An alignment consists of six lines: From top to bottom: Predicted secondary structure for the target (mother stem in blue), target sequence, template sequence, native secondary structure for the template (mother stem in blue), annotated sequence for the template (realised mother stem in blue, realised child stems in red), and realised secondary structure for the template (mother stem in blue, child stems in red).
4.2. TESTING OF FRAMEWORK COMPONENTS

introduced into the target sequence.

The results suggest that \( n_X \) should not be assigned a fixed global value. Rather, the value \( n_X \) should depend on the length difference of both sequences. As the alignment algorithm is based on the global Needleman-Wunsch sequence alignment, the length of the shorter template fragments should be increased to match or even slightly surpass the length of the target. Otherwise, the shorter fragment might become scattered along the target due to the nature of the global alignment. Setting \( n_X \) to 100 leads to alignment scattering (not depicted).

Finding multiloop-fragments in different tRNA structures

This test attempts at finding tRNA-multiloops in different tRNA targets (cf. Fig. 4.6). With the first alignment we try to find a multiloop-fragment from the E-chain of a tRNA analogue from the ribosome complex (PDB:486d) in the target sequence. The sequence of the template fragment is highly similar to the target sequence and the alignment yields a similar result as the previous intra-2tra alignments.

We perform a second alignment to find the 2tra-multiloop in another tRNA target (PDB:1ehz). Target and template show relatively low sequence similarity. Also, while the template contains a 3-multiloop, the target contains only two child stems both of which include short internal loops. The only structure which is realised by the alignment is the mother stem.

Finding a hairpin fragment in its source structure

In earlier experiments [49] the Lara algorithm was not capable of aligning sequences of significantly different lengths. Aligning a typical hairpin sequence to a long sequence would result in the introduction of a large number of gaps into the hairpin sequence (scattering). To determine whether Lara is capable to correctly align a short sequence against a long structure, we perform an alignment of two hairpin sequences against their source structure (PDB:2tra). The resulting alignments show the shorter sequences aligned as a whole to the 5’-end of their source structure instead of aligning them to their expected positions within the structure. Adjusting the gap-penalty values does not yield better results.

The results show that the mother stem can apparently be modeled completely by trying to match a complete larger interior loop fragment. A separated modeling approach could be promising. It could be attempted to model the outmost (small) hairpin fragments first and consequently just align the remaining body to a library of the larger fragments.
CHAPTER 4. COMPUTATIONAL RESULTS

Figure 4.5: Dotplots: Predicted and Verified Secondary Structures for 2tra
A dotplot is an \( n \times n \)-matrix, displaying the probability for two bases from the same sequence \( S \) of length \( n \) to form a base-pair. The cell \((i, j)\) is filled with a black square corresponding to the squared pairing-probability for the bases \( S_i, S_j \). Left: Prediction by RNAfold (with probabilities \( p \in [0,1] \)). Right: Secondary structure computed with RNAVIEW from experimentally verified data (with probabilities \( p \in \{0,1\}\)).

Figure 4.6: tRNA Test Alignments. Alignments for 1ehz and 486d with \( n_X = 10 \), or 7, respectively. An alignment consists of six lines. From top to bottom: Predicted secondary structure for the target (mother stem in blue), target sequence, template sequence, native secondary structure for the template (mother stem in blue), annotated sequence for the template (realised mother stem in blue, realised child stems in red), and realised secondary structure for the template (mother stem in blue, child stems in red).
4.2. TESTING OF FRAMEWORK COMPONENTS

4.2.2 Isostericity Matrix Validation

This experiment targets at the validation of the applicability of isostericity matrices as a scoring scheme or base-pair validation method in RNA threading. A valid scoring scheme/validation method must be capable of recognising the target structure within a set of alignments resulting from sliding a target sequence along a reference structure which is identical to the native structure of the target sequence: let $S = s_1, \ldots, s_n$ be an RNA sequence of length $n$ and $C = c_1, \ldots, c_n$ be the secondary and tertiary structure of $S$. $C$ is now shifted along $S$ one position at a time (gapless threading), starting with $c_1$ aligned to $s_n$ ($c_1 \rightarrow s_n$, $c_n \rightarrow \emptyset$, $s_1 \rightarrow \emptyset$) and both $c_n$ and $s_1$ unaligned ($\emptyset$). Each of the resulting alignments $a_1, \ldots, a_n$ is then processed by the contact validation routine, testing each pair of bases $(s_i, s_j)$ that is aligned to a contact $(c_i, c_j)$ of the contact-map. If $(s_i, s_j) = (c_i, c_j)$ or if $(s_i, s_j)$ can form a base-pair that is isosteric to $(c_i, c_j)$, then the contact $(c_i, c_j)$ can be realised by bases $s_i, s_j$. For each alignment a contact map overlap score $q$ is then computed:

$$ q = \frac{r}{n}, $$

where $r$ denotes the number of secondary and tertiary structure contacts realised by the alignment and $n$ denotes the total number of contacts included within the alignment (i.e. a contact $(b_i, b_j)$ is included in an alignment if both bases $b_i$ and $b_j$ are part of the aligned subsequence). A recognition of $C$ within $S$ occurs, when the complete alignment $a_n$ ($c_1 \rightarrow s_1$, $c_n \rightarrow s_n$) yields a $q$-score of $q_n = 1.0$ and all other alignments $a_i$ yield a $q$-score $q_i < q_n$: mapping each base of the structure to its corresponding base within the sequence is the only valid solution when attempting to recognise a structure within its sequence. An alignment $a_x$ with $q_x = q_n$ implies that $x = n$, otherwise there would exist a substructure within $C$ which would be falsely recognised as the original structure.

The examined contact map (PDB:2tra) contains 35 contacts (20 secondary structure base-pairs and 15 (tertiary) contacts). The sequence length is 74. As described above, we expect that the complete alignment $a_{74}$ yields a $q$-score of 1.0. We also expect that all contacts can be realised within $a_{74}$. Figures 4.7 and 4.8 display the results for the alignments. These expectations are met: all 35 contacts are realised within $a_{74}$, yielding a $q$-score of $q_{74} = 1.0$. The curves depicting the numbers of unaligned contacts/base-pairs show "platforms", i.e. regions where the current number of unaligned contacts/base-pairs stays constant. The length of the transitions between two of these platforms is related to the length of the stem that ends at the first position of the transition region. Furthermore, it can be observed that there are no realised base-pairs and/or contacts until the alignment starts at position 58 (contacts) or respectively 51 (base-pairs). This can be explained by the facts that firstly, no contacts exist after position 58 and secondly, the last secondary structure base-pair is located at (51,59) (cf. Fig. 4.9 for reference).

We perform a second set of gapless threadings (cf. also Figures 4.7 and 4.8), aligning the contact
map of 2tra against its reversed ribonucleotide sequence. The structure is not symmetric and thus we expect a deviation from the forward score. As certain types of base-pairs are not isosteric to their reversed pair (e.g. G-U cis Watson-Crick are not isosteric to U-G cis Watson-Crick pairs), it is possible that some base-pairs will or will not be recognised – even in a symmetric structure. The best scoring alignment is $a_5$ ($q_5 = 0.46$). This can be explained by the fact that the four residues on the 3’-end of the 2tra sequence are unpaired (dangling) ends. Although, there is a possibility that any of the four dangling residues can form base-pairs with other bases which actually would match base-pairs of the main stem, the actual secondary structure starts after the first four residues of the reverse sequence.
4.2. TESTING OF FRAMEWORK COMPONENTS

Figure 4.7: Gapless Threading of 2tra - Realised Contacts

Figure 4.8: Gapless Threading of 2tra - Unaligned Contacts
4.3 Semi-automatic modeling of selected structures

To determine if the fragment-based threading method can generally be applied to create model structures for a given RNA sequence, we perform two proof-of-concept experiments. These aim at building the three-dimensional structure of two target molecules (PDB:2tra, 1u9s) using fragments of different RNA molecules. The experiments are conducted in two steps. First, we identify the different structure motifs included in the target structure (e.g. the tRNA consists of one 3-multiloop, one single hairpin (the anticodon-loop), and a kissing hairpin bi-loop motif (cf. Fig. 4.10)) and the search fragment library for template candidates displaying a similar secondary structure as the target motif. An alignment of candidate and target is then submitted to a superposition routine together with the respective atom coordinates. We then compute an optimal superposition of the backbone atoms \((P, O_5^*, O_3^*, C_5^*, C_3^*, C_4^*)\) of the target and template molecules, employing the method of Kabsch [23]. The algorithm computes a rigid transformation to create an optimal matching of two point sets \(v, w\), with \(|v| = |w|\), minimising their root mean square deviation (RMSD):

\[
RMSD(v, w) = \sqrt{\frac{1}{|v|} \sum_{i=1}^{|v|} ||v_i - w_i||^2}
\]

Thus, the Kabsch algorithm simulates the work of a model-assembly process, allowing an estimation of the general applicability of the fragment-based threading approach independent of the employed sequence-to-structure alignment method.

During the second step, the transformed atom coordinates of the template structure are superpo-
4.3. SEMI-AUTOMATIC MODELING OF SELECTED STRUCTURES

Positioned with the corresponding region within the target structure and visualised by the molecular editing and visualisation software amiraMol [26].

4.3.1 Proof of concept: tRNA

For the modeling of a tRNA structure (PDB: 2tra) we use a library of 274 hairpins, twelve kissing hairpins, and 23 3-multiloops. The hairpins, kissing hairpins, and five multiloops are selected from the same secondary structure fold class as their respective reference counterparts. The remaining 18 multiloops are chosen from a different fold class.

Table 4.1 shows RMSD-intervals for superpositions of the hairpin-templates with the native target structure: 65 superpositions show an RMSD $\geq 4\AA$ and $< 5\AA$, 143 have an RMSD $\geq 5\AA$ and $< 6\AA$, 18 have an RMSD $\geq 6\AA$ and $< 7\AA$, and 10 templates display RMS-deviations $\geq 7\AA$ and $< 8\AA$. The RMS-values of the remaining 38 templates deviate more than 8\AA from the native structure of the target. Tables 4.2 and 4.3 show the RMS-deviations for the superpositions of multiloop and kissing-hairpin templates with the native structure of the target. Nine kissing-hairpin templates show an RMSD of $\leq 2.5\AA$, the remaining three deviate more than 2.5\AA from the native structure of the target (cf. Table 4.2). Only five out of the 23 multiloop templates show RMSD values $\leq 5\AA$ (cf. Table 4.3). Three templates (3tra:1 - 0.42440\AA, 486d:11 - 0.42442\AA, 486d:1 - 2.29679\AA) deviate less than 2.5\AA from the target structure. For the assembled structure model cf. Fig. 4.10. To demonstrate the possibility of substituting a certain structure motif with a fragment which belongs to a different fold class, we do not assemble the structure from the three fragments yielding the lowest RMS deviation (PDB: 3tra:1, 486d:19, 2csx:6). Instead we replace the 3tra:1-multiloop with the best scoring fragment (PDB: 1ttt:1) from a different secondary structure fold class. Templates for the anticodon hairpin (PDB: 2csx:6) and main multiloop (PDB: 1ttt:1) are tRNA fragments while the kissing hairpin (PDB: 486d:19) belongs to a tRNA-analogue from the ribosome complex.

The results of the semi-automatic modeling do not turn out as expected. The hairpin modeling all yield an RMSD $\geq 4\AA$ (cf. Table 4.1). This might indicate that hairpin structures are not as simple to model despite their rather short sequence lengths. An interesting observation can be made regarding the distribution of RNA species among the individual RMSD intervals: The majority (71%) of the hairpin structures yielding RMSD-values between 4\AA and 5\AA belong to source molecules that are classified as rRNA by their PDB data. (cf. Fig. 4.11). Despite the fact of the actual numbers of fragments from each species-type (tRNA:161, rRNA:107), members of the same species as the reference structure (tRNA) only constitute 24% of the hairpins of the [4, 5]-interval. In contrast to the hairpin structures, superposing of the kissing hairpin fragments results in RMSD values between 1.4\AA and 2.9\AA. As kissing hairpins consist of two single hairpins the difference between the results is rather surprising. A possible reason for the difficulty of modeling the single hairpin loop could be that the target is in fact the anti-codon loop. This could indicate that the anti-codon loops of different tRNAs might adopt individual three-dimensional structures.
which could facilitate the recognition of their respective codon on an mRNA.

The five multiloop fragments of the same fold class as the reference structure all yield RMSD values between 0.4Å and 2.75Å while the remaining 18 multiloops yield RMSD values $> 5\text{Å}$.

<table>
<thead>
<tr>
<th>Hairpin Fragments</th>
<th>RMSD[Å]</th>
<th>#Fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[4, 5]</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>[5, 6]</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>[6, 7]</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>[7, 8]</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>$[8, \infty]$</td>
<td>38</td>
</tr>
</tbody>
</table>

Figure 4.10: Assembly of a tRNA model structure. Left: Reference secondary structure of the target (PDB:2tra) and contained secondary structure motifs. In parentheses: Fragment-ID of the fragment used to model a respective motif. Right: Superposition of the sugar-phosphate backbones of fragments and reference structure.
### 4.3. Semi-automatic Modeling of Selected Structures

#### Kissing Hairpin Fragments

<table>
<thead>
<tr>
<th>Fragment</th>
<th>486d:19</th>
<th>3tra:6</th>
<th>486d:20</th>
<th>1f7v:6</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>1.40918</td>
<td>1.40921</td>
<td>1.40925</td>
<td>1.66011</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>1f7u:6</th>
<th>lll2:14</th>
<th>lasz:13</th>
<th>lasz:11</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>1.69186</td>
<td>1.75928</td>
<td>1.99788</td>
<td>2.32182</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>lasy:13</th>
<th>lll2:12</th>
<th>2byt:14</th>
<th>2bte:14</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>2.4203</td>
<td>2.73008</td>
<td>2.76543</td>
<td>2.85903</td>
</tr>
</tbody>
</table>

Table 4.2: Semi-automatic Modeling of 2tra: Kissing Hairpins.

#### Multiloop Fragments

<table>
<thead>
<tr>
<th>Fragment</th>
<th>3tra:1</th>
<th>486d:11</th>
<th>486d:1</th>
<th>lasy:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>0.42440</td>
<td>0.42442</td>
<td>2.29679</td>
<td>2.58715</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>lasz:1</th>
<th>lttt:1</th>
<th>1fcw:17</th>
<th>4tra:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>2.72258</td>
<td>5.16406</td>
<td>5.61439</td>
<td>5.61458</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>1fcw:9</th>
<th>1fcw:5</th>
<th>1fcw:1</th>
<th>1fcw:13</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>5.6149</td>
<td>5.61491</td>
<td>5.61494</td>
<td>5.61494</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>1evv:1</th>
<th>6tna:1</th>
<th>1ehz:1</th>
<th>1j9v:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>5.64289</td>
<td>5.6652</td>
<td>5.67653</td>
<td>6.13837</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>1sz1:1</th>
<th>1sz1:5</th>
<th>1gix:1</th>
<th>1jgo:1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Fragment</th>
<th>1jgp:1</th>
<th>1jgpq:1</th>
<th>1ttt:9</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSD[Å]</td>
<td>24.33004</td>
<td>24.33004</td>
<td>29.54355</td>
</tr>
</tbody>
</table>

Table 4.3: Semi-automatic Modeling of 2tra: Multiloops.
CHAPTER 4. COMPUTATIONAL RESULTS

4.3.2 Proof of concept: A-Type S Domain of RNAse P

The second experiment aims at modeling an A-type S domain of the ribozyme Ribonuclease P (RNAse P). The A- and B-type S domains of RNAse P illustrate an interesting example of structural diversity in homologous RNAs: while differing significantly in secondary and tertiary structure the A- and B-types of the S domain have a similar three-dimensional structure [27]. This means that the three-dimensional structure of an RNA molecule can in fact be built up from different secondary and tertiary structure fragments. Being able to completely or even just partially assemble such an example structure using fragments from completely different types of RNAs would prove the applicability of the fragment based threading approach.

Figure 4.12 displays the native structure (PDB:1u9s) of the A-type domain. It juxtaposes three-dimensional and secondary structure. The molecule is partitioned into eight secondary structure motifs (P7-P14). The coordinate file for the domain structure does not include an actual unpaired
region for the P12 hairpin-loop, nevertheless the length of that region (4 residues) is known. Figure 4.13 depicts the model structure resulting by semi-automatically performed superposition of different structure fragments with the target structure (PDB:1u9s). Nine motifs (seven rRNA, two RNAse P, cf. Table 4.4) could be modeled successfully, constituting 60% of the complete native structure (without the loop region of the P12 hairpin). The majority of the motifs that could not be modeled correspond either to loop regions or to the P12 motif. It has been attempted to find fitting template fragments for this motif, but the respective candidates did not yield results with an RMSD < 8Å. Nevertheless, the coverage of a large part of the target structure, using structure fragments from different template molecules, shows that a target RNA can at least partially be assembled of structure fragments originating from different RNA molecules. The missing loop regions suggest an extension of the template library to additionally include unpaired regions.
<table>
<thead>
<tr>
<th>Motif</th>
<th>Fragment</th>
<th>RMSD [Å]</th>
<th>RNA Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEM1 (P11')</td>
<td>1fka:121</td>
<td>0.56615</td>
<td>rRNA</td>
</tr>
<tr>
<td>STEM2 (P12')</td>
<td>1fka:121</td>
<td>2.44232</td>
<td>rRNA</td>
</tr>
<tr>
<td>STEM3 (P14')</td>
<td>1pbr:3</td>
<td>1.34993</td>
<td>rRNA</td>
</tr>
<tr>
<td>MLOOP1 (P7')</td>
<td>1j5e:84</td>
<td>0.56022</td>
<td>rRNA</td>
</tr>
<tr>
<td>ILOOP1 (P13')</td>
<td>1z58:108</td>
<td>0.8272</td>
<td>rRNA</td>
</tr>
<tr>
<td>HPIN1 (P8')</td>
<td>2a2e:9</td>
<td>1.62336</td>
<td>RNAse P</td>
</tr>
<tr>
<td>HPIN2 (P9')</td>
<td>2h0p:88</td>
<td>2.89099</td>
<td>rRNA</td>
</tr>
<tr>
<td>HPIN3 (P13')</td>
<td>1k01:169</td>
<td>1.84996</td>
<td>rRNA</td>
</tr>
<tr>
<td>HPIN4 (P14')</td>
<td>1xsu:2</td>
<td>1.79529</td>
<td>RNAse P</td>
</tr>
</tbody>
</table>

Table 4.4: Semi-automatic modeling of 1u9s: Template Fragments

Figure 4.12: Native Structure for the A-Type S Domain of RNAse P. Top: Secondary structure for the A-type domain. Bottom: Three-dimensional structure with annotated secondary structure motifs (P7-P14).
Figure 4.13: Model Structure for the A-Type S Domain of RNAse P. Nine structure fragments of distinct templates cover 60% of the native structure of the A-Type S domain. Top: Secondary structure for the A-type domain. Unmodeled motifs are marked by asterisks. Bottom: Three-dimensional structure with annotated secondary structure motifs (P7′-P14′). Distinct modeled are marked by different colours. Grey: Hairpin loop part of hairpin templates. Only fragments yielding an RMSD < 4Å from a superposition with the target structure have been used.
4.4 RNA Design: Hairpin Modeling

One area of application for RNA structure prediction and modeling is the field of RNA Design, i.e. the artificial synthesis of RNA structure elements for application in therapeutical and medical research environments (cf. e.g. Jaeger et al. [15]). A possible application within the field of RNA design could be the modification of structure elements of a given target molecule, aiming at a functional change of the target. For instance, it could be attempted to modify the structure of an aptamer such that the binding affinity towards a certain ligand is increased (or decreased). The hairpin modeling experiments target at the search of suitable replacement structures for available hairpin loops with known structure.

4.4.1 Modeling of the P12 Hairpin of the A-Type S Domain of RNAse P

As described in Sect. 4.3.2, the P12 motif of the A-Type S Domain of RNAse P is not completely included in the provided atom coordinate resource (PDB:1u9s). Finding possible candidate structures to model the missing motif is a step towards the field of RNA Design. The P12 motif is a 4-hairpin with a stem region consisting of 32 bases, including three bulges (the missing part is marked by asterisks):

```
123456789012345678901234567890123456
((((..(((((((((****))))))).)))))..))
```

Due to its length and the included bulges, the complete motif is rather unique within the template library. Thus, we only model the un-bulged part (residues 9-26) of P12, which can theoretically be regarded as a simple 4-hairpin with a stem of length 7. The sequence of P12 (GCCGGGGCUUCCCCCGGU) and its secondary structure (((((((((****)))))))))) are submitted to the threading framework and aligned against a hairpin library containing 5948 hairpins, taking roughly one hour of computation time on a 1.66GHz machine with 1GB of RAM. In addition to the score resulting from the Lara threading ($l$-score), we score the resulting alignments via isostericity matrix validation (cf. Sect.4.2.2). A contact map overlap score

$$q = \frac{r}{n}$$

is computed, where $r$ denotes the number of secondary structure contacts realised by the alignment and $n$ denotes the total number of secondary structure contacts included in the alignment. The best $l$-scoring alignment (39.24 - template 1n32:65) only achieves a $q$-score of 0.57. For the further processing we regard only candidates which result in alignments with $q = 1.0$ (i.e. all contacts within the template structure can be formed by the target sequence) and which display...
the same loop size as the target data. This yields a total number of 33 candidate structures for the P12 motif which all show the same \( l \)-score of 33.1 and sequence identity (GCGUCCGAAAGGCACG)U. The RMSD values for superposing the candidate structures with the corresponding motif within the target molecule are then computed in the same way as in Sect. 4.2.2. All superpositions show RMSD values between 0.807 Å and 0.873 Å. The superposition of template 1yhq:61 with the target yields the lowest RMSD (0.807 Å, 1n32:65 yields 2.14095 Å). Fig. 4.14 depicts the superposition of the partial P12 motif with templates 1yhq:61 and 1n32:65.

Figure 4.14: Superposition of the P12 Hairpin with Template Hairpins. Dark purple: partial P12 motif of an A-type S domain of RNAse P (PDB:1u9s), yellow: template yielding lowest RMSD (PDB:1yhq), blue: template yielding the best \( l \)-score (PDB:1n32).

The framework-supported search for suitable template structures for a part of the P12 hairpin of the A-type S domain of RNAse P together with the preceding semi-automatic modeling experiments allow a 70% coverage of the complete three-dimensional structure of the A-type S domain of RNAse P with local RMS deviations not larger than 3 Å. The experiment suggests a further breakdown of the template structures into smaller fragments, e.g. to divide stem regions of certain lengths to allow the coverage of a wider range of target structures.
4.4.2 Modeling of a HIV-1 A-rich Hairpin Loop

This experiment attempts at finding possible replacement structures for a viral RNA hairpin loop (PDB:1bvj). For the target sequence (GGCGACGGUGUAACUCCG) we predict a secondary structure using a secondary structure prediction tool (((((((.((......)))))))))). The target is then processed by the threading framework: it is aligned against a library containing 5948 hairpins. The complete threading takes approximately 1.5 hours of computation time. The results are filtered by their q-score: alignments scoring less than q = 0.5 are filtered, yielding a set of 2241 candidate structures for the structure model of the target. Superposing these candidates with the native structure of the target results in a set of 38 candidates showing an RMSD less than 3.0Å. The template with the lowest RMSD (1.936Å) is a hairpin from a tRNA transcript (PDB:2byt:9). The candidate cannot cover the loop region of the target. Superposing the template yielding the best l-score (PDB:1k6h:1, l-score: 46.779) with the target structure results in an RMSD of 2.8Å. Fig. 4.15 depicts the superposition of the native structure of the target with templates 2byt:9 and 1k6h:1. The stem regions of the target and template structures follow a highly similar backbone trace. The loop regions of 1k6h:1 and target differ significantly, giving an explanation for the rather high RMSD of 1k6h:1.

Figure 4.15: Superposition of an HIV-1 A-rich Hairpin with Template Hairpins. Dark purple: HIV-1 A-rich Hairpin (PDB:1bvj), yellow: template yielding lowest RMSD (PDB:2byt), blue: template yielding the best l-score (PDB:1k6h).
4.5 Modeling of tRNA Structures

This experiment aims at building model structures for tRNA targets utilising the threading framework. The secondary structure of a tRNA molecule consists of three or four different secondary structure motifs: a multiloop, the anticodon hairpin, and the D-loop and T-loop hairpins. The D-loop and T-loop hairpins can be regarded as either a single tertiary structure motif (kissing hairpins) or as two individual hairpins. We perform separate modelings for each motif type, using a different library for each modeling: a multiloop library containing 726 templates, a hairpin library consisting of 5948 templates, and a kissing hairpin library containing 272 templates. The computing time for aligning a single target against a complete library varies between 15 minutes and 2.5 hours. For the hairpin alignment, we employ hairpin detection to find possible regions for a hairpin within the target sequence (cf. Sect. 3.3). During the threading process it could be observed that the alignments of kissing hairpins against the target sequence does not work properly: the introduction of the separator sequences as proposed in Sect. 4.2.1 leads to the misalignment of both hairpin parts of the motif to start and end regions of the target sequence. The alignments resulting from kissing hairpin alignments are discarded and we attempt to cover both the D-loop and the T-loop with hairpin alignments. While the T-loop region of the target is detected without trouble, the D-loop region can not be found by the hairpin detection. Thus, we cannot model the D-loop for the examined targets. The assembly of the different fragments is performed in three steps. The first step is to find a feasible multiloop fragment. We attempt to find a fragment which originates from a different RNA family than the target. However, the best scoring fragments are tRNA multiloops. Hence, it is at least attempted to find a template originating from a different tRNA type. The next step is choosing a hairpin fragment that can cover the anticodon hairpin. The last step is the search for templates that might cover the remaining two D-loop and T-loop hairpins. All hairpin alignments and the multiloop alignment are submitted to the assembly routine described in Sect. 3.5. The resulting cliques are then filtered: only cliques that contain both multiloop and anticodon hairpin template are regarded as a valid suggestion for the final structure model.

For the modeling of the tRNA(asp) 2tra, we use a multiloop of a tRNA(asp) (PDB:1asy:3, RMSD 2.584 Å), a hairpin originating from 18S rRNA (PDB:1uuu:3, RMSD 4.837 Å), and a tRNA hairpin (PDB:2ab4:1, RMSD 7.119 Å). The rRNA hairpin serves as template to model the anticodon hairpin. While the template for the anticodon hairpin motifs fits into one of the missing regions not covered by the multiloop template, the T-loop template does not. This observation can be made during all tRNA modelings: while the anticodon loop can usually be modeled easily, alignments with the T-loop are always gapped and show q-scores less or equal than 0.5. D-loop alignments are in general never aligned to the D-loop region directly but instead are part of a long alignment starting at the 5'-end of the target sequence. Figure 4.16 depicts the model structure for 2tra. We model the structure for the tRNA(phe) 6tna using a tRNA(thr) multiloop template (PDB:1qf6:1, RMSD 2.772 Å), a 16S rRNA hairpin (PDB:2aw7:65,
RMSD 8.674 Å), and a tRNA(phe) anticodon-hairpin (PDB:1ehz:3, RMSD 1.657 Å). Figure 4.17 depicts the model structure for 6tna. For the model structure of the tRNA(tyr) 1j1u a tRNA(phe) multiloop (PDB:1f9v:1, 2.251 Å) is used. The anticodon hairpin is modeled based on a tRNA(cys) anticodon (PDB:1b23:5, RMSD 1.797 Å). For modeling the T-loop, a hairpin originating from 23S rRNA (PDB:2aar:158, RMSD 3.369 Å) is employed. Figure 4.18 depicts the model structure for 1j1u.

<table>
<thead>
<tr>
<th>Motif</th>
<th>Fragment</th>
<th>RMSD[Å]</th>
<th>RNA Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>2tra:tRNA(asp)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiloop</td>
<td>1asy:3</td>
<td>2.584</td>
<td>tRNA(asp)</td>
</tr>
<tr>
<td>Anticodon Hairpin</td>
<td>1uuu:3</td>
<td>4.837</td>
<td>18S rRNA</td>
</tr>
<tr>
<td>T-loop hairpin</td>
<td>2ab4:1</td>
<td>7.119</td>
<td>tRNA</td>
</tr>
<tr>
<td>6tnat:tRNA(phe)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiloop</td>
<td>1q61:1</td>
<td>2.772</td>
<td>tRNA(thr)</td>
</tr>
<tr>
<td>Anticodon Hairpin</td>
<td>1ehz:3</td>
<td>1.657</td>
<td>tRNA(phe)</td>
</tr>
<tr>
<td>T-loop hairpin</td>
<td>2aw7:65</td>
<td>8.674</td>
<td>16S rRNA</td>
</tr>
<tr>
<td>1j1u:tRNA(tyr)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiloop</td>
<td>1f9v:1</td>
<td>2.251</td>
<td>tRNA(phe)</td>
</tr>
<tr>
<td>Anticodon Hairpin</td>
<td>1b23:5</td>
<td>1.797</td>
<td>tRNA(cys)</td>
</tr>
<tr>
<td>T-loop hairpin</td>
<td>2aar:158</td>
<td>3.369</td>
<td>23S rRNA</td>
</tr>
</tbody>
</table>

Table 4.5: Modeling of tRNA Structures: Template Fragments
4.5. MODELING OF tRNA STRUCTURES

Figure 4.16: Model Structure for 2tra. Left: Alignment of the template secondary structures against the target sequence. Asterisks: internal gaps in the template, dashes: end-gaps, braces: non-realised secondary structure contacts, parentheses: realised secondary structure contacts Right: Backbone superposition of the native target structure and template structures. Colours: Red: multiloop structure, Blue: anticodon hairpin, Yellow: T-loop hairpin, Grey: target structure

Figure 4.17: Model Structure for 6tna. Left: Alignment of the template secondary structures against the target sequence. Asterisks: internal gaps in the template, dashes: end-gaps, braces: non-realised secondary structure contacts, parentheses: realised secondary structure contacts Right: Backbone superposition of the native target structure and template structures. Colours: Red: multiloop structure, Blue: anticodon hairpin, Yellow: T-loop hairpin, Grey: target structure
Chapter 5

Conclusion

In this thesis a new approach for fragment-based RNA structure modeling has been proposed. A description of the composition of a template structure library and the design and implementation of the single components for a threading framework as well as the evaluation of the utilised methods and algorithms has been given. We have demonstrated the fragment-based modeling of RNA three-dimensional structure on selected RNA structure examples: we were able to successfully model partial tRNA structures as well as roughly 70% of the three-dimensional structure of an A-type S domain of the RNase P complex. The results of the conducted experiments show that in general a given RNA structure can be assembled, using structure fragments originating from an RNA of a different species. This leads to the conclusion that a fragment based approach can indeed be applied to the RNA structure modeling problem and as such may serve as a promising method in the fields of RNA structure prediction and RNA Design. There are still unanswered questions (e.g. how to assemble the final three dimensional structure or how to determine the best structure model) for building a complete structure model for an arbitrary RNA sequence. However, the capability of determining alternative structure fragments to replace existing structure motifs within a target structure might prove useful for application in RNA Design, e.g. modeling of target binding sites in aptamers and riboswitches.

5.1 Problem Discussion

While the fragment-based approach is in general applicable for the RNA structure prediction problem, the completely automatic prediction of RNA three-dimensional structure is not yet possible. Starting from the generation of the template fragment library and ending at the assembly of different feasible template fragments to a complete model structure, there is a whole range of issues to be solved.
5.1.1 The Template Library

The atom coordinates serving as the foundation for the contact maps stored in the template fragment library, originate from the PDB. Due to the nature of the structure elucidating experiments via X-ray crystallography and NMR, a PDB database entry does not necessarily contain a complete molecule. Missing data varies from single atoms to entire ranges of residues, depending on the main goals of the respective experiment. The lack of certain structural features, such as the loop region of the RNAse P A-type S domain (cf. Sect. 4.3) disturbs the functioning of the fragment detection routines described in Chapter 2 and thus may lead to incorrect fragment data. In fact, these incomplete structures together with the difficulties that occur while processing pseudoknotted structures, constitute the main reason for the failure of the motif detection routines (cf. Sect. 2.4.1 and Sect. 2.4.2). Template molecules including pseudoknots present a difficult challenge for the partition of a template secondary structure into distinct motifs. Due to the many different definitions of pseudoknots (e.g. [4, 16, 54]) it is almost impossible to find a solution that satisfies the majority of occurring cases.

The current motif detection routines operate on a structure graph, building the secondary structure from all cis Watson/Crick base-pairs and disregard any other base-pair or contact while classifying the secondary structure motifs. It might be helpful, if the restrictive definition of RNA secondary structure is weakened or even disregarded during motif detection. Regardless of their actual base-pair family, any non-crossing base-pair could be regarded as belonging to the secondary structure, any crossing base-pair would be classified as pseudoknot-inducing. This would also prevent the occasionally occurring "holes" (induced by the replacement of a "real" secondary-structure base-pair with a base-pair of another type) within the stem regions of the current fragment library.

In addition, the residue numbering of a PDB entry does not always follow a uniform scheme. Despite the fact that it has been attempted to solve this index problem by re-numbering the residues in a preprocessing step, there are still candidate molecules that cannot be processed properly due to an unordinary numbering scheme.

Another problem that occurred during the base-pair detection step is the unrated base-pair detection performed by RNAVIEW: the method allows the occurrence of multiple base-pair contacts via the same edge for a given base, i.e., the method would detect the formation of base-pairs with two different bases via the Watson-Crick edge of a single base and classify both resulting pairs as canonical Watson-Crick pairs, which is sterically impossible. While this problem can simply be solved by applying a filter prior to evaluating the results of RNAVIEW, it raises doubts about the overall quality of the method.
5.2. OUTLOOK

5.1.2 RNA Structure Alignment

In a majority of cases the RNA structure alignment performed by Lara does not align regions that are actually expected to be aligned. Examples are given in Sect. 4.2.1. Basically, the failure to find and align the proper regions, can be accredited to two major reasons. First, the secondary structure prediction that is submitted to the Lara algorithm is only an insufficient estimation of the actual secondary structure of the target, not even including possible tertiary structure contacts. The usage of base-pair probabilities instead of actually predicted structures, produces a slight relief, as they contain all possible contacts for the target. However, even the usage of base-pair probabilities does not lead to much better results. The introduction of a feasible tertiary contact prediction could possibly improve the alignment.

The second reason for the failure to align desired regions to each other are the occurrences of dangling ends (unpaired regions at either terminal of a ribonucleotide sequence) in the actual structure of the target. The sequence alignment part of Lara might falsely align template stem-regions to these unpaired regions. While it is easy to avoid misalignments due to dangling ends in a test setting (by removal of the respective subsequences), it is virtually impossible to do so in an actual application to an unknown target sequence. The ability to access any (suboptimal) alignment resulting from a single Lara run might yield a number of candidate alignments without template regions misaligned to dangling end regions of the target.

To be capable of regarding all tertiary contacts within an RNA structure, the alignment should allow multiple contacts per base (cf. Sect. 1.1.1). The current version of the Lara algorithm only allows single contacts and thus can only include a subset of the available contacts into the alignment score. The disregarded contacts may still be realised by the alignment. However, as they do not contribute to the final score, it might be possible that the alignment with the actual optimal template (i.e. the template that represents the best fitting three-dimensional structure) scores less than another alignment with a less optimal template.

5.2 Outlook

5.2.1 Extension and Refinement of the Template Library

The described problems with the template data suggest a refinement of the library, to increase the number of templates available for the modeling of target structures. Also, the missing loop regions from the RNase P A-type S domain modeling experiment suggest the employment of an additional fragment-type, i.e. unpaired regions.

5.2.2 Selecting Template Candidates for RNA Design

RNA Design experiments can benefit from computational planning such as the automatic pre-selection of template candidates. To be applicable in such an environment, the template candidate selection process needs to be refined to attain a higher degree of user-friendliness. Currently,
template candidates are chosen via a filtering script that simply compares their lengths to the length of the target or subregions of the target. While this method is convenient to somewhat limit the size of the search space and thus lessen the overall running time, it might exclude suitable candidates or include structurally improper templates.

A more reasonable approach might be the utilisation of an RNA-structure-specific grammar, describing the desired template secondary and/or tertiary structure. For RNA secondary structure, such a grammar could be easily implemented using regular expressions operating on the respective bracket notation. To enable the incorporation of tertiary structure and secondary structure pseudoknotted regions, another solution has to be found: due to the crossing base-pairs included in these structures, it is not possible to operate on the bracket notation alone. A solution to this problem could be offered by graph representation of the secondary (as proposed by Pasquali et al. [43]) and tertiary structure. Another approach could be a possible extension of the dynamic programming method for computing common sequential and structural patterns between two RNAs as proposed by Backofen and Siebert [3].

5.2.3 Isostericity Matrices

The introduction of isostericity matrices (IMs) to the RNA structure modeling problem is a promising step for the validation and evaluation of resulting structure models. The inclusion of these matrices into the scoring function of a structural alignment algorithm such as Lara would allow a more specific selection of the residues which are to be aligned. For the Lara algorithm the integration of isostericity matrices would imply the exclusion of alignment edges which map two non-isosteric base-pairs onto each other. For secondary structure alone, the validation guaranteed by the IMs is not of high importance as the majority of secondary structure pairs (A-U and G-C) are isosteric and thus exchangeable anyway. However, as soon as the alignment method is extended to include tertiary structure information as well, such a validation is critical as two non-isosteric base-pairs simply cannot be swapped without altering the three-dimensional structure of the molecule, possibly inducing a change or even loss of function as worst case.

5.2.4 Fragment Assembly

When searching for template candidates to model a target structure, the best fitting structure fragments have to be assembled to form a three-dimensional structure model for the given target sequence. This assembly problem can be divided into two sub-problems. The first sub-problem (two-dimensional) is to decide which fragment candidates are assembled to form a structure model as modeling runs for a target structure yield fragments for different regions of the target. RNA secondary structure motifs are overlapping (cf. Sect. 1.1.2), hence overlaps between distinct fragments will inevitably occur during the assembly. The problem can be modeled similar to the chaining performed in sequence assembly or by performing clique detection on a graph represent-
5.2. OUTLOOK

ing the different fragments. The complexity increases for each additional region to be modeled. For instance, a tRNA molecule consists of three or four motifs which have to be covered by the modeling runs, as such the cliques of interest are only of size three or four.

The second sub-problem (three-dimensional) is the spatial assembly of the actual atom coordinates, i.e. finding the optimal rigid transformations that bring the different parts of the model structure together. Such a problem could be attempted to solve using semi-automatic modeling methods as described in Sect. 1.2 (e.g. MC-SYM) or other methods from the field of optimisation such as Simulated Annealing (cf. e.g. [50]).

While a general approach for the two-dimensional problem has been proposed in Sect. 3.5, finding possible detailed solutions for both sub-problems is beyond the scope of this thesis but hopefully provides motivation for future work in this direction.
Appendix A

Definitions

- **Definition: Secondary Structure Motif.** A secondary structure motif is defined as a single stem (mother stem), the following unpaired residues (loop region), and all outgoing stems (child stems) of the loop region.

- **Definition: Source Structure.** A source structure is the complete native secondary structure of an RNA molecule. It consists of a set of overlapping secondary structure motifs.

- **Definition: Secondary Structure Fragment.** A secondary structure fragment is defined as a secondary structure motif which has been clipped from its source structure.

- **Definition: Tertiary Structure Motif.** A tertiary structure motif is a set of secondary structure fragments which are connected by at least one tertiary contact. A tertiary structure motif consisting of exactly two fragments can be called a tertiary bi-motif.

- **Definition: (Subsequence) Region.** A region is a sequence of nucleobases.

- **Definition: Stem Region.** A stem region is a set of consecutive base-pairs (stem) and optional unpaired regions (bulges).

- **Definition: Target (Sequence).** A ribonucleotide sequence of which a model structure is to be predicted. (Also: Query Sequence)

- **Definition: Target Structure.** A target structure is the native structure of the target sequence. By comparison of target structure and model structures resulting from a modeling approach, the overall predictive quality of the approach is determined.
Appendix B

Library Contents

The source molecules of the template fragment library are listed by their PDB id. The numbers in parentheses display the amount of fragments that have been found in the respective molecules.

<table>
<thead>
<tr>
<th>PDB id</th>
<th>Number of Fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td>17ra</td>
<td>2</td>
</tr>
<tr>
<td>1a1t</td>
<td>3</td>
</tr>
<tr>
<td>1a4t</td>
<td>2</td>
</tr>
<tr>
<td>1a51</td>
<td>3</td>
</tr>
<tr>
<td>1a60</td>
<td>5</td>
</tr>
<tr>
<td>1a9l</td>
<td>2</td>
</tr>
<tr>
<td>1a9n</td>
<td>3</td>
</tr>
<tr>
<td>1afx</td>
<td>2</td>
</tr>
<tr>
<td>1ajf</td>
<td>2</td>
</tr>
<tr>
<td>1aju</td>
<td>2</td>
</tr>
<tr>
<td>1akx</td>
<td>2</td>
</tr>
<tr>
<td>1am0</td>
<td>3</td>
</tr>
<tr>
<td>1anr</td>
<td>3</td>
</tr>
<tr>
<td>1aq0</td>
<td>3</td>
</tr>
<tr>
<td>1arj</td>
<td>2</td>
</tr>
<tr>
<td>1asy</td>
<td>18</td>
</tr>
<tr>
<td>1asz</td>
<td>15</td>
</tr>
<tr>
<td>1ato</td>
<td>2</td>
</tr>
<tr>
<td>1atv</td>
<td>2</td>
</tr>
<tr>
<td>1atw</td>
<td>2</td>
</tr>
<tr>
<td>1b23</td>
<td>10</td>
</tr>
<tr>
<td>1b36</td>
<td>4</td>
</tr>
<tr>
<td>1bau</td>
<td>4</td>
</tr>
<tr>
<td>1bgz</td>
<td>2</td>
</tr>
<tr>
<td>1biv</td>
<td>2</td>
</tr>
<tr>
<td>1bn0</td>
<td>2</td>
</tr>
<tr>
<td>1bj2</td>
<td>4</td>
</tr>
<tr>
<td>1bjy</td>
<td>2</td>
</tr>
<tr>
<td>1byj</td>
<td>4</td>
</tr>
<tr>
<td>1bx2</td>
<td>2</td>
</tr>
<tr>
<td>1bx3</td>
<td>3</td>
</tr>
<tr>
<td>1bzt</td>
<td>2</td>
</tr>
<tr>
<td>1ck8</td>
<td>3</td>
</tr>
<tr>
<td>1cn8</td>
<td>3</td>
</tr>
<tr>
<td>1cn9</td>
<td>3</td>
</tr>
<tr>
<td>1cq5</td>
<td>4</td>
</tr>
<tr>
<td>1cql</td>
<td>4</td>
</tr>
<tr>
<td>1cx0</td>
<td>8</td>
</tr>
<tr>
<td>1d0t</td>
<td>2</td>
</tr>
<tr>
<td>1d0u</td>
<td>2</td>
</tr>
<tr>
<td>1d6k</td>
<td>3</td>
</tr>
<tr>
<td>1ddy</td>
<td>39</td>
</tr>
<tr>
<td>1dk1</td>
<td>5</td>
</tr>
<tr>
<td>1drz</td>
<td>8</td>
</tr>
<tr>
<td>1dul</td>
<td>4</td>
</tr>
<tr>
<td>1e4p</td>
<td>3</td>
</tr>
<tr>
<td>1e7k</td>
<td>5</td>
</tr>
<tr>
<td>1e8o</td>
<td>7</td>
</tr>
<tr>
<td>1e95</td>
<td>4</td>
</tr>
<tr>
<td>1ebq</td>
<td>3</td>
</tr>
<tr>
<td>1ebr</td>
<td>3</td>
</tr>
<tr>
<td>1ebs</td>
<td>3</td>
</tr>
<tr>
<td>1ec6</td>
<td>3</td>
</tr>
<tr>
<td>1efw</td>
<td>15</td>
</tr>
<tr>
<td>1eht</td>
<td>6</td>
</tr>
<tr>
<td>1ehz</td>
<td>7</td>
</tr>
<tr>
<td>1et2</td>
<td>2</td>
</tr>
<tr>
<td>1eti</td>
<td>10</td>
</tr>
<tr>
<td>1ekz</td>
<td>2</td>
</tr>
<tr>
<td>1eor</td>
<td>3</td>
</tr>
<tr>
<td>1esh</td>
<td>2</td>
</tr>
<tr>
<td>1esy</td>
<td>3</td>
</tr>
<tr>
<td>1et4</td>
<td>50</td>
</tr>
<tr>
<td>1etf</td>
<td>3</td>
</tr>
<tr>
<td>1etg</td>
<td>3</td>
</tr>
<tr>
<td>1eqa</td>
<td>10</td>
</tr>
<tr>
<td>1euy</td>
<td>10</td>
</tr>
<tr>
<td>1evv</td>
<td>7</td>
</tr>
<tr>
<td>1exd</td>
<td>11</td>
</tr>
<tr>
<td>1exy</td>
<td>2</td>
</tr>
<tr>
<td>1fft</td>
<td>3</td>
</tr>
<tr>
<td>1f27</td>
<td>2</td>
</tr>
<tr>
<td>1f5u</td>
<td>4</td>
</tr>
<tr>
<td>1f6u</td>
<td>3</td>
</tr>
<tr>
<td>1f6x</td>
<td>2</td>
</tr>
<tr>
<td>1f6z</td>
<td>2</td>
</tr>
<tr>
<td>1f78</td>
<td>2</td>
</tr>
<tr>
<td>1f79</td>
<td>2</td>
</tr>
<tr>
<td>1f7f</td>
<td>2</td>
</tr>
<tr>
<td>1f7g</td>
<td>2</td>
</tr>
<tr>
<td>1f7h</td>
<td>2</td>
</tr>
<tr>
<td>1f7i</td>
<td>2</td>
</tr>
<tr>
<td>1f7u</td>
<td>7</td>
</tr>
<tr>
<td>1f7v</td>
<td>7</td>
</tr>
<tr>
<td>1f7y</td>
<td>5</td>
</tr>
<tr>
<td>1f84</td>
<td>3</td>
</tr>
<tr>
<td>1f85</td>
<td>2</td>
</tr>
<tr>
<td>1f91</td>
<td>3</td>
</tr>
<tr>
<td>1fcw</td>
<td>46</td>
</tr>
<tr>
<td>1feq</td>
<td>2</td>
</tr>
<tr>
<td>1fk</td>
<td>274</td>
</tr>
<tr>
<td>1ffy</td>
<td>7</td>
</tr>
<tr>
<td>1ffz</td>
<td>53</td>
</tr>
<tr>
<td>1fg0</td>
<td>57</td>
</tr>
<tr>
<td>1fhk</td>
<td>2</td>
</tr>
<tr>
<td>1fir</td>
<td>9</td>
</tr>
<tr>
<td>1fje</td>
<td>2</td>
</tr>
<tr>
<td>1fjg</td>
<td>152</td>
</tr>
<tr>
<td>1fka</td>
<td>158</td>
</tr>
<tr>
<td>1fh8</td>
<td>2</td>
</tr>
<tr>
<td>1fnn</td>
<td>3</td>
</tr>
<tr>
<td>1foq</td>
<td>26</td>
</tr>
<tr>
<td>1fqz</td>
<td>3</td>
</tr>
<tr>
<td>1fyo</td>
<td>4</td>
</tr>
<tr>
<td>1fyp</td>
<td>4</td>
</tr>
<tr>
<td>1g1x</td>
<td>5</td>
</tr>
<tr>
<td>1g59</td>
<td>13</td>
</tr>
<tr>
<td>1g7o</td>
<td>3</td>
</tr>
<tr>
<td>1gax</td>
<td>17</td>
</tr>
<tr>
<td>1gid</td>
<td>23</td>
</tr>
<tr>
<td>1gix</td>
<td>21</td>
</tr>
<tr>
<td>1gkv</td>
<td>3</td>
</tr>
<tr>
<td>1gkw</td>
<td>3</td>
</tr>
<tr>
<td>1grz</td>
<td>52</td>
</tr>
<tr>
<td>1grt</td>
<td>7</td>
</tr>
<tr>
<td>1gts</td>
<td>7</td>
</tr>
<tr>
<td>1h3e</td>
<td>10</td>
</tr>
<tr>
<td>1h4q</td>
<td>7</td>
</tr>
<tr>
<td>1h4s</td>
<td>7</td>
</tr>
<tr>
<td>1hec</td>
<td>15</td>
</tr>
<tr>
<td>1hiji</td>
<td>2</td>
</tr>
<tr>
<td>1hix</td>
<td>2</td>
</tr>
<tr>
<td>1hnw</td>
<td>147</td>
</tr>
<tr>
<td>1hm</td>
<td>147</td>
</tr>
<tr>
<td>1hns</td>
<td>155</td>
</tr>
<tr>
<td>1ho6</td>
<td>2</td>
</tr>
<tr>
<td>1hoq</td>
<td>2</td>
</tr>
<tr>
<td>1hq1</td>
<td>4</td>
</tr>
<tr>
<td>1hr0</td>
<td>143</td>
</tr>
<tr>
<td>1hr2</td>
<td>27</td>
</tr>
<tr>
<td>1hs1</td>
<td>2</td>
</tr>
<tr>
<td>1hs2</td>
<td>2</td>
</tr>
<tr>
<td>1hs3</td>
<td>2</td>
</tr>
<tr>
<td>1hs4</td>
<td>2</td>
</tr>
<tr>
<td>1hs8</td>
<td>2</td>
</tr>
<tr>
<td>1hvu</td>
<td>13</td>
</tr>
<tr>
<td>1hwq</td>
<td>3</td>
</tr>
<tr>
<td>1i3x</td>
<td>2</td>
</tr>
<tr>
<td>1i3y</td>
<td>2</td>
</tr>
<tr>
<td>1i46</td>
<td>3</td>
</tr>
<tr>
<td>1i4b</td>
<td>2</td>
</tr>
<tr>
<td>1i4c</td>
<td>2</td>
</tr>
<tr>
<td>1i6u</td>
<td>3</td>
</tr>
<tr>
<td>1i94</td>
<td>156</td>
</tr>
</tbody>
</table>
APPENDIX B. LIBRARY CONTENTS

| 1h95 (160) | 1l96 (153) | 1i97 (149) | 1i9f (4) | 1i9v (7) | 1ibk (154) | 1ibl (153) |
| 1ibm (149) | 1idv (2) | 1ie1 (3) | 1ie2 (3) | 1il1 (2) | 1il2 (3) | 1iil2 (15) |
| 1ivs (18) | 1ljh (7) | 1ljb (12) | 1lj4y (2) | 1lj5a (287) | 1lj5e (152) | 1ljbr (3) |
| 1ljbt (5) | 1lgp (21) | 1lgq (21) | 1ljid (3) | 1lj2 (267) | 1lj07 (6) |
| 1ljox (2) | 1ljp0 (2) | 1ljtj (2) | 1ljtw (2) | 1lj7 (2) | 1ljur (2) | 1ljwc (4) |
| 1ljzc (2) | 1ljx (286) | 1ljzy (286) | 1ljzz (288) | 1lk01 (284) | 1lk2g (3) | 1lk4a (2) |
| 1lk5b (2) | 1lk6g (2) | 1lk6h (2) | 1lk73 (267) | 1lk8a (275) | 1lk8w (2) |
| 1lk9m (269) | 1lk9w (7) | 1lkaj (4) | 1lkc8 (274) | 1lkdl (271) | 1lk6 (5) | 1lkis (4) |
| 1lka (2) | 1lks (3) | 1lkog (18) | 1lkos (3) | 1lp7 (3) | 1lkp (4) | 1lpvy (4) |
| 1lkpz (4) | 1lkq5 (269) | 1lkuo (3) | 1lkup (5) | 1lkxk (3) | 1llc (3) | 1llw (3) |
| 1llx (4) | 1ll3d (4) | 1ll8v (23) | 1ll9a (13) | 1llce (3) | 1lzd (3) | 1ln (11) |
| 1lu3 (2) | 1luu (2) | 1lux (2) | 1lvc2 (2) | 1lm1k (267) | 1lm5k (9) | 1lm5l (5) |
| 1lm50 (8) | 1lm5p (9) | 1lm5v (9) | 1lm82 (3) | 1lm90 (268) | 1lme0 (2) | 1lme1 (2) |
| 1lmfj (3) | 1lmf (2) | 1lmf (2) | 1lmf (3) | 1lmj (15) | 1lmj (3) | 1lmj (7) |
| 1lmm (17) | 1lnm (3) | 1lnm (3) | 1lns (3) | 1lmt4 (2) | 1lmz (7) | 1ln (156) |
| 1ln33 (165) | 1ln34 (159) | 1ln36 (179) | 1ln66 (3) | 1ln77 (13) | 1ln78 (13) | 1ln8 (270) |
| 1ln8x (3) | 1lna2 (3) | 1lnb (2) | 1lnb (2) | 1lnb (26) | 1ln0 (2) | 1lnem (2) |
| 1lnj (268) | 1lnj (263) | 1lnj (262) | 1lnjo (263) | 1lnjp (263) | 1lnkw (286) | 1lnwx (285) |
| 1lnwy (286) | 1lnyb (2) | 1lnyi (2) | 1lnz1 (3) | 1ln0b (9) | 1ln0c (9) | 1ln0d (6) |
| 1lob2 (7) | 1lob5 (31) | 1loln (9) | 1lond (272) | 1lona (5) | 1loq (2) | 1losw (3) |
| 1low9 (3) | 1lp5m (4) | 1lp5m (5) | 1lp5m (6) | 1lp5p (6) | 1lp6v (4) | 1lp9x (287) |
| 1lpr (4) | 1lpwy (2) | 1lpm (93) | 1lpnu (200) | 1lpns (84) | 1lpny (202) | 1lpq (29) |
| 1lpq (2) | 1lpw (272) | 1lpw (273) | 1lq82 (274) | 1lq86 (268) | 1lq8n (5) | 1lq93 (8) |
| 1lpq6 (9) | 1lqa (3) | 1lqa (15) | 1lq6 (2) | 1lq6 (2) | 1lq6 (7) | 1lqf (2) |
| 1lqs (9) | 1lqrt (9) | 1lqrt (9) | 1lqt (9) | 1lqu (2) | 1lqv (273) | 1lqv (270) |
| 1lqwa (2) | 1lqwb (2) | 1lqzw (13) | 1lt2p (3) | 1lt3e (2) | 1lt7w (2) | 1lt7 (2) |
| 1lraw (3) | 1lrf (3) | 1lrf (2) | 1lrk (2) | 1lrl (7) | 1lrnm (4) | 1lrng (2) |
| 1lmk (4) | 1lroq (2) | 1lry1 (19) | 1ls0 (5) | 1ls1h (142) | 1lsli (269) | 1ls2f (3) |
| 1ls (2) | 1ls (268) | 1ls (6) | 1scl (3) | 1ser (13) | 1sj3 (8) | 1sj4 (8) |
| 1ls (8) | 1ls (2) | 1slp (3) | 1sm (278) | 1sly (3) | 1szy (3) | 1s3 (13) |
| 1lsy (2) | 1lt28 (3) | 1lt4l (2) | 1ltbk (2) | 1ltfn (2) | 1ltje (3) | 1lir (3) |
| 1ltm2 (9) | 1ltob (3) | 1lta (9) | 1lttt (19) | 1ltx (3) | 1lub (10) | 1lub (2) |
| 1lu3k (2) | 1lu6 (7) | 1lu6b (7) | 1lu6p (5) | 1lu8d (12) | 1lu9 (23) | 1ull (3) |
| 1lum (12) | 1lus (2) | 1lud (2) | 1lui (2) | 1luu (2) | 1lubx (8) | 1luby (8) |
| 1lvb (8) | 1vco (8) | 1vco (8) | 1vco (8) | 1vco (8) | 1vco (5) | 1vop (2) |
| 1vq (105) | 1vor (201) | 1vos (105) | 1vou (104) | 1vou (99) | 1vow (198) | 1vow (105) |
| 1vq (193) | 1vog (99) | 1vpg (99) | 1vq (271) | 1vq (265) | 1vq (268) | 1vq (269) |
Appendix C

Parameters

C.1 Hairpin Detection

For the hairpin detection (Sect. 3.3) we mostly use the default parameters. The window size $w$ is computed basing on the length of the template $t$: $w = t - (t \% 10) + 10$. Additionally, an offset $o = 5$ has been added. The halfstem-length corresponds to the halfstem-length of the template. We set the minimum halfstem-length to 1 and the gap penalty to -1.

C.2 Lara

We run Lara with two parameter sets, based on the default parameters: the "short" set is employed if the sequence lengths of target and template differ by more than 20 bases. This set uses a generator suboptimality of 300 compared to the default value of 60: in certain cases using the default value caused the implementation to crash, e.g. during the alignment of a hairpin (20-30 residues) to a target sequence of length 70. Editing the gap penalties does not yield better results, so we keep the default values. We modify the sequence scoring matrices such that aligning any base against a separator $\bar{X}$ (cf. Sect. 3.2) yields a match score of 0.5. The length $n_X$ of the separator sequences corresponds to the length difference of target and template.
Bibliography


