COMPUTATIONAL SEARCH OF RNA PSEUDOKNOTS AND STRUCTURAL VARIATIONS IN GENOMES

by

ZHIBIN HUANG

(Under the Direction of Liming Cai)

ABSTRACT

Non-coding RNA (ncRNA) secondary structural homologs can be detected effectively in genomes based on a covariance model (CM) and associated dynamic programming algorithms. However, the computational difficulty in aligning an RNA sequence to a pseudoknot structure has prohibited high throughput search for RNA pseudoknot structures in sequences. Due to the lack of appropriate ncRNA structural evolution models, accurate search of distant RNA structural homologs also remains difficult.

The core of both problems is the sequence structure alignment that requires intensive computation for complex structure. Based on a conformational graph model we built to incorporate all the interactions of stem and loop, including the crossing stem pattern of pseudoknots, the sequence-structure alignment problem can be modeled as a subgraph isomorphism problem. Based on the graph tree decomposition and naturally small tree width in ncRNA structures including pseudoknots, the problem of searching ncRNA with pseudoknot structures in genomes can be solved efficiently by the tree decomposition based dynamic programming algorithm. Further, the sequence-structure alignment problem for distant RNA structural homolog search can be modeled as a graph homomorphism problem. Tree

decomposition based dynamic programming algorithm equipped with the new technique of NULL stem is applied to solving the RNA structural variation search problem more effectively.

In this dissertation, we developed two search frameworks, RNATOPS and its extension RNAv, based on a general conformational graph model. Our genome search test results demonstrate RNATOPS has an advantage over Infernal and other methods in accuracy and computational efficiency when searching for the ncRNA pseudoknot structures in genomes, and RNAv, with the capability of detecting pseudoknot, also has an advantage over Infernal in detection of some distant homologs.

INDEX WORDS: ncRNA structure search, Sequence structure alignment, pseudoknot, ncRNA structure variation, RNA structure evolution, Graph tree decomposition, Dynamic Programming

COMPUTATIONAL SEARCH OF RNA PSEUDOKNOTS AND STRUCTURAL VARIATIONS IN GENOMES

by

ZHIBIN HUANG

B.S., Huaqiao University, Fujian, China, 2000

M.S., Huaqiao University, Fujian, China, 2003

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2011

COMPUTATIONAL SEARCH OF RNA PSEUDOKNOTS AND STRUCTURAL VARIATIONS IN GENOMES

by

ZHIBIN HUANG

Major Professor:	Liming Cai
Committee:	Russell Malmberg
	Ying Xu
	Khaled Rasheed

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2011

DEDICATION

I dedicate this dissertation to my families and friends who have supported me during the time of my study in UGA.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr.Liming Cai for his continuous guidance during my graduate study at University of Georgia, Dr. Russell Malmberg for introducing and teaching me the fundamental knowledge in biological science and also providing me the data and his guidance to conduct my research in bioinformatics, Dr. Ying Xu and Dr. Khaled Rasheed for serving in my dissertation committee and the valuable suggestions and help from them. Last, but far from least, I would like to thank all of my friends in the RNA-Informatics Lab and Computer Science Department at UGA.

TABLE OF CONTENTS

Page

CHAPTER 1

INTRODUCTION

1.1 COMPUTATIONAL RNA GENE FINDING

RNA genes are genes that do not encode proteins; they are also called non-coding RNA genes (ncRNAs) [1]. NcRNAs have been shown to be involved in many biological processes including gene regulation, chromosome replication and RNA modification [2, 3, 4]. Recent biological studies [1, 5, 37] have indicated that there may be thousands types of ncRNAs.

There are two main kinds of computational methods to detect RNA genes: *ab initio* method and comparative analysis method.

1.1.1 *AB INITIO* GENE FINDING

In spite of many years of studies in RNA gene finding, progress has been limited. An *ab initio* method finds encoded RNA genes from genome sequences without annotations. It can be classified into the following two main frameworks: the compositional information method and the minimum free energy (MFE) based method.

The Compositional information method assumes that ncRNAs have on average a GC content of 50%, which works when searching for GC-rich islands in some AT-rich organisms [6]. But this assumption may not always be true when it is applied to search for new RNA genes in other organisms.

The MFE based method is a long established paradigm and it has been used to search for RNA genes [36]. This method is based on the following assumptions: (1) At equilibrium, the solution to the underlying molecular folding problem is unique, i.e., the molecule folds into the

lowest energy state; (2) the free energies of individual structural motifs are additive [7]. However, the structure corresponds to the MFE may not be the correct one: the MFE structure may not always be the one adopted by the RNA [8]. The correct structure may be among those with sub-optimal free energies [9]. In particular, the comprehensive analysis by Gardner and Giegerich [10] shows that comparative methods tended to systematically outperform MFE methods in predicting the RNA genes.

As pointed out by Rivas and Eddy [11], secondary structure prediction on a single sequence is insufficient to reliably predict ncRNA genes. We now introduce the more powerful method - comparative analysis method.

1.1.2 COMPARATIVE ANALYSIS GENE FINDING

Given two or more related species, alignments of their whole genomes can be performed to find the maximum regions of similarity between them. For multiple genomes alignment, these can be built by comparing two genomes at a time then building a multiple alignment by extracting aligned regions common to all genomes. QRNA[12], RNAZ[13] and EVOFOLD[14] are programs which provide a measure of probability that a given alignment of sequences adopts a conserved RNA fold [37].

1.1.3 STRUCTURAL PROFILE BASED METHOD

The structural profile based method mainly works for ncRNA annotations [37] and it can be built based upon the comparative analysis method. From an alignment of multiple sequences with annotated structures, a structural *profile* can be built, which contains the statistical information of the consensus structure and sequence of the annotated ncRNA sequences. Searching in the genomes to compare sequence segments against this profile mainly relies on a procedure called

sequence-structure alignment. A high alignment score indicates the corresponding sequence segment has a high probability of containing the structure pattern of the given ncRNA.

The number of possible alignments between a sequence and a structure profile grows exponentially in the sequence length. The alignment result is usually the one with the highest alignment score, thus the task is actually an optimization problem. The Covariance Model (CM) was introduced by Eddy and Durbin [15] to profile RNA secondary structure. The optimal sequence-structure alignment between a sequence and a pseudoknot-free structure profiled with CM can be accomplished with the CYK dynamic programming algorithm in time $O(WN^3)$, where W is the size of the model and N is the sequence length. Due to such time complexity needed for the sequence-structure alignment, CM-based search may not be efficient enough on complex or large RNA structures and long genomes, with resorting to sophisticated speed up techniques.

A pseudoknot is a special RNA structure, which has a stem-loop where the loop pairs with another region of the RNA. Because of the crossing stem pattern in a pseudoknot structure, which CM can not profile, pseudoknot structure prediction problem is NP-hard [20, 21]. Proposed pseudoknot profiling models [22-26] are mostly extensions of CM. However, the time and space complexities for optimal sequence structure alignment based on these models are $O(N^4)$ or $O(N^5)$, which is not practical for efficient search of pseudoknot structures.

1.2 STRUCTURAL VARIATION SEARCH

The structural search tools have been most successful in identifying ncRNAs homologs with little or small structural variation between sequences. RNA secondary and tertiary structures are both constant and variable across evolution [23-25]; that is, some sub-structures, such as stem-loops, will be found in all members of a given ncRNA family, but other sub-structures will be

found only in some of the sequences of the family. Such structural variations pose novel challenges in profiling distant homologs for effective searches [25]. In previous work [26-30] RNA base and base pair evolution information was incorporated into SCFG models. An improved model for RNA structural evolution has also been proposed [23, 31] which can deal with limited degree of structure rearrangement between homologs but has yet to be incorporated into a search program. To date, a general method that addresses both possible misalignments and structural variation is still missing [32]. Searches for structurally distant homologs still largely rely on customized methods or tools [24].

1.3 CONTRIBUTION OF THE DISSERTATION

In this dissertation, we propose methods to address the two challenging problems that have been identified above: (1) efficient search for ncRNA with pseudoknot structures in genomes and (2) accurate search for ncRNA with structural variations in genomes.

The core of both problems is the sequence-structure alignment. First, a conformational graph model [33, 34] was built to incorporate all interactions of stems and loops including the crossing stems in pseudoknot structures. Then the sequence structure alignment problem is modeled as a subgraph isomorphism problem. Due to the naturally small tree widths in ncRNA structures including pseudoknot, searching for ncRNA pseudoknot structure in genomes can be solved efficiently by a tree decomposition-based dynamic programming algorithm with graph isomorphism. Further, the sequence-structure alignment problem for distant RNA structural homology search is modeled as a graph homomorphism problem. The RNA structural variation search problem is solved by the tree decomposition based dynamic programming algorithm equipped with new techniques based on the notion of NULL stem.

In this dissertation, we developed two search frameworks, RNATOPS [34] and its extension RNAv [35], based on the general conformational graph model. Genome search test results show that RNATOPS' performance is comparable with or better than the state-of-the-art algorithm, Infernal, in identifying large or complex RNA structures including pseudoknots, while taking a much smaller amount of time. Test results on 51 benchmark data set used by Infernal show that RNAv has the capability of detecting pseudoknots with a comparable performance to the latest version of Infernal and performs better in detecting some distant homologs.

1.4 DISSERTATION OUTLINE

This dissertation is organized as follows. Chapter 2 introduces some fundamental terms used in this dissertation. Chapter 3 presents a detailed survey of the computational techniques, graph tree decomposition, and related dynamic programming algorithms for optimization problems. In Chapter 4 and 5, we describe in detail about how to develop this type of algorithms to solve ncRNA pseudoknot search and ncRNA structural variation search. Test results of these algorithms in genome search will also be presented. Chapter 6 concludes the dissertation and provides discussions for possible future research.

1.5 REFERENCES

[1] Meyer, I. M. 2007. A practical guide to the art of RNA gene prediction. Brief. Bioinform. 8:396–414.

[2] Frank,D.N. and Pace,N.R. (1998) Ribonuclease P: unity and diversity in a tRNA processing ribozyme. Annu. Rev. Biochem., 67, 153–180.

[3] Nguyen,V.T. et al. (2001) 7SK small nuclear RNA binds to and inhibits the activity of CDK9/cyclin T complexes. Nature, 414, 322–325.

[4] Yang,Z. et al. (2001) The 7SK small nuclear RNA inhibits the CDK9/cyclin T1 kinase to control transcription. Nature, 414, 317–322.

[5] S. Gisela. An expanding universe of noncoding RNAs. Science. 2002. 296:1260-1263.

[6] R. J. Klein, Z. Misulovin, and S. R. Eddy, "Noncoding RNA genes identified in AT-rich hyperthermophiles", Proc. Natl. Acad. Sci. USA, vol. 99, pp. 7542-7547, 2002.

[7] Y. Ding, Statistical and Bayesian approaches to RNA secondary structure prediction, RNA,, vol. 12, pp. 323-331, 2006.

[8] D. H. Mathews, Revolutions in RNA secondary structure prediction, J. Mol Biol, vol. 359, pp. 526–532, 2006.

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

[10] PP Gardner, R Giegerich, A comprehensive comparison of comparative RNA structure prediction approaches. BMC Bioinformatics, 5, (2004) 140.

[11] Rivas E, Eddy SR. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. Bioinformatics 2000;16(7):583–605.

[12] Rivas E, Eddy SR. Noncoding RNA gene detection using comparative sequence analysis. BMC Bioinformatics 2001;2:8.

[13] Washietl S, Hofacker IL, Stadler PF, Fast and reliable prediction of noncoding RNAs. Proc Natl Acad Sci USA 2005;102:2454–59

[14] Pedersen JS, Bejerano G, Siepel A, et al. Identification and classification of conserved RNA secondary structures in the human genome. PLoS Comput Biol 2006;2:e33.

[15] Eddy,S.R. and Durbin,R., 1994, RNA sequence analysis using covariance models. Nucleic Acids Res., 22, 2079–2088.

[16] Akutsu T, 2000, Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. Discrete Applied Mathematics, 104:45-62.

[17] Lyngso, R. and Pedersen, C. 2000. RNA pseudoknot prediction in energy-based models, J. Comput. Biol., Vol. 7, pp. 409-427.

[18] Uemura,Y. et al. 1999. Tree adjoining grammars for RNA structure prediction. Theoret. Comput. Sci., 210, 277–303.

[19] Rivas E, Eddy S, 1999, Pknots A dynamic programming algorithm for RNA structure prediction including pseudoknots. J Mol Biol, 285(5):2053-2068.

[20] Cai L, et al., 2003, Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics 19(Suppl 1):i66-i73.

[21] Hiroshi Matsui, 2005, Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures.

[22] Yuki Kato, et al. 2007, RNA Pseudoknotted Structure Prediction Using Stochastic Multiple Context-Free Grammar.

[23] Holmes I: A probabilistic model for the evolution of RNA structure. BMC Bioinformatics 2004, 5:166.

[24] Axel Mosig. et al. (2009) Customized strategies for discovering distant ncRNA homologs, Briefings in Functional Genomics and Proteomics, doi:10.1093/bfgp/elp035.

[25] Menzel P, Gorodkin J, Stadler PF. (2009) The tedious task of finding homologous non-coding RNA genes. RNA, 15(12):2075-2082, 2009.

[26] Knudsen B, Hein J (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics 15: 446–454.

[27] Knudsen B, Hein J (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. Nucleic Acids Res 31: 3423–3428.

[28] Pedersen JS, Meyer IM, Forsberg R, Simmonds P, Hein J (2004a) A comparative method for finding and folding RNA secondary structures within protein-coding regions. Nucleic Acids Res 32: 4925–4936.

[29] Pedersen JS, Forsberg R, Meyer IM, Hein J (2004b) An evolutionary model for protein-coding regions with conserved RNA structure. Mol Biol Evol 21: 1913–1922.

[30] Pedersen, J.S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E., Rogers, J., Kent, J., Miller, W., and Haussler, D. Identification and Classification of Conserved RNA Secondary Structures in the Human Genome. PLoS Computat Biol 2(4), e33 (2006).

[31] Bradley RK, Holmes I (2009) Evolutionary Triplet Models of Structured RNA. PLoS Comput Biol 5(8): e1000483. doi:10.1371/journal.pcbi.1000483.

[32] Andreas R. Gruber, et al. 2010. Rnaz 2.0: Improved Noncoding Rna Detection, Pacific Symposium on Biocomputing 15:69-79.

[33] Song,Y. et al. (2005) Tree decomposition based fast searching for RNA structures with and without pseudoknots. Proc. IEEE Comput. Syst. Bioinform. Conf., IEEE Computer Society Press. 223–234.

[34] Huang, Z., Y. Wu, J. Robertson, L. Feng, R. Malmberg, and L. Cai. 2008. Fast and accurate search for non-coding rna pseudoknot structures in genomes. Bioinformatics. 24:2281–2287.

[35] Huang,Z., Malmberg,R., Mohebbi, M and Cai,L. 2010. RNAv: Non-coding RNA Secondary Structure Variation Search via Graph Homomorphism, In Proceedings of Computational Systems Bioinformatics Conference (CSB 2010), August, 2010. Vol. 9, p. 56-69.

[36] Edvardsson,S., et al., 2003, A search for H/ACA snoRNAs in yeast using MFE secondary structure prediction. Bioinformatics, 19, 865–873.

[37] S. Griffiths-Jones, Annotating noncoding RNA genes, Annu. Rev. Genomics Hum. Genet. 8 (2007), pp. 279–298.

CHAPTER 2

FUNDAMENTALS

This chapter introduces some fundamental terms that will be used often throughout this dissertation.

2.1 RNA SECONDARY STRUCTURE

RNA is a single-stranded molecule consisting of four nucleotides adenine(A), cytosine(C), guanine(G) and uracil(U) [1], and typically folds onto itself to form a secondary structure through Watson-Crick base-pairs (G-C, A-U) and wobble (G-U) base-pairs [24, 25]. Generally it is believed that RNA secondary structure is conserved across related species but the contiguous nucleotides in a sequence can change significantly [1]. Base pairs are approximately coplanar [26] and are almost always stacked each other forming so-called *stems*. (Fig. 2.1). Other than stems, elements of an RNA secondary structure include *loops*, which are non-paired subsequences enclosed by base pairs. Single stranded bases occurring within a stem are called a *bulge* if the single stranded bases are on only one side of the stem; A loop at the end of a stem is called a *hairpin loop* [1] if it is short.

2.2 PSEUDOKNOT AND ITS SEARCH IN GENOMES

A *pseudoknot* (Figure. 2.1) is an RNA structure that consists of at least two stems in which half of one stem is intercalated between the two halves of another stem.

Pseudoknot structures constitute only a minority in current structural databases [2], however, this may be more a reflection of the difficulty to detect them than their true abundance in nature. As pseudoknots are known to play diverse and important functional roles in biology [3], it would

be good to invest efforts into developing novel algorithms that can model pseudoknot structures in a conceptually more elegant and computationally more efficient way [2].

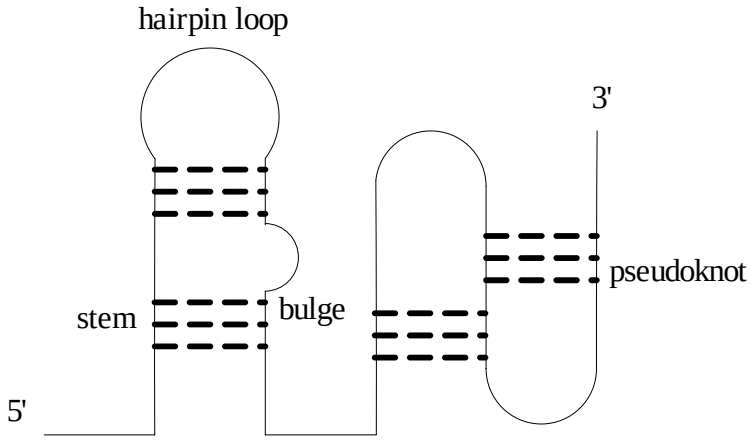


Figure. 2.1 A RNA secondary structure

Different stochastic models [4-8] were proposed to model the RNA pseudoknot structures, and they are listed in Table 2.1. Because of the intensive computation of sequence structure alignment based on these models, they have not been implemented for efficient searching.

Table 2.1 RNA pseudoknot structures modeling methods

Stochastic model	Time complexity (N is the number of nucleotides)
Tree adjoining grammars [4]	$O(N^4)$ for simple pseudoknots; $O(N^5)$ or more for the other pseudoknots
Pknots [5]	$O(N^6)$
Parallel communicating grammar systems (PCGS) [6]	$O(N^6)$
Pair Stochastic Tree Adjoining Grammars (PSTAG) [7]	$O(N^5)$
Stochastic Multiple Context-Free Grammar (SMCFG) [8]	$O(N^5)$

Intersecting CMs have also been proposed for modeling pseudoknots [9], and used to search small genomes [10], yet resulting in the same efficiency issue. Heuristic search methods [11, 27] have been developed that can work with RNAs containing pseudoknots, but with other

limitations. For example, ERPIN [11], considers the individual stem loops contained in a secondary structure. It scans a genome to find possible hit locations for each stem loop. A hit for the overall structure is reported when there exists a combination of hit locations for individual stem loops that conform to the overall structure. One disadvantage is that ERPIN does not allow gaps in the alignment and thus may have low sensitivity for targets that are remote homologs of the query structure model.

Another strategy to search for a ncRNA pseudoknot structure in genome is by removing the crossing stems of this ncRNA pseudoknot structure, resulting in a structure without pseudoknot, that can be identified by search tools like Infernal. But the strategy may not produce correct results [12]. So far, there are no efficient ncRNA pseudoknot structure search tools readily to be used. How to efficiently search for ncRNA containing pseudoknot structures in genomes is one of the challenges in RNA bioinformatics.

2.3 RNA STRUCTURAL HOMOLOG SEARCH

Homologous RNAs have common ancestry and are expected to have similar structures and some similarity in sequence [13-14]. Structural-homology is about a conserved structure from a set of related RNA sequences, which can be described as a consensus structure across these molecules.

Homology-based searching methods have become important for annotation of ncRNAs [15]. Algorithms to search for homologs based on RNA structures can be divided into two classes: a) specific to a particular RNA class, for example, tRNAscanSE [16] for tRNAs; b) general enough applicable to all structured RNAs, for example, INFERNAL [17] and ERPIN [11].

Currently the most successful general approach for detecting structural homologs of known ncRNAS is based on the covariance model (CM), a special type of stochastic context-free grammar (SCFG), introduced by Eddy and Durbin [18]. The CM can profile position-specific

compensatory mutations between base pairs as well as base conservations, yielding accurate ncRNA specific and reconfigurable structural homolog search tools.

2.4 RNA STRUCTURAL VARIATION

RNA structural variation can be classified into the following three types [23]. Type 1: single base insertion, deletion and substitution; Type 2: base-pair insertions, deletions and substitutions within a conserved stem, and Type 3: insertion and deletion of the entire secondary structure elements.

Figure 2.2 gives a structure alignment of the nanos translational control element (TCE) from *Drosophila*; it illustrates these three types of RNA structural variations: Type 1 single base deletion can be found in the loop region between the right arm of stem 2 and the left arm of stem 3, DVU24695 has sequence “UUA” while DRONANOS does not have any sequence; Type 2 base-pair deletion and substitution can be found in stem 2 region. DVU24695 has stem 2 with length of 6 while DRONANOS has stem 2 with length 9; also in DVU24695, one base-pair A-U is substituted with G-U in DRONANOS; Type 3 stem insertion/deletion can be found in stem 0 of DVU24695 and DRONANOS.

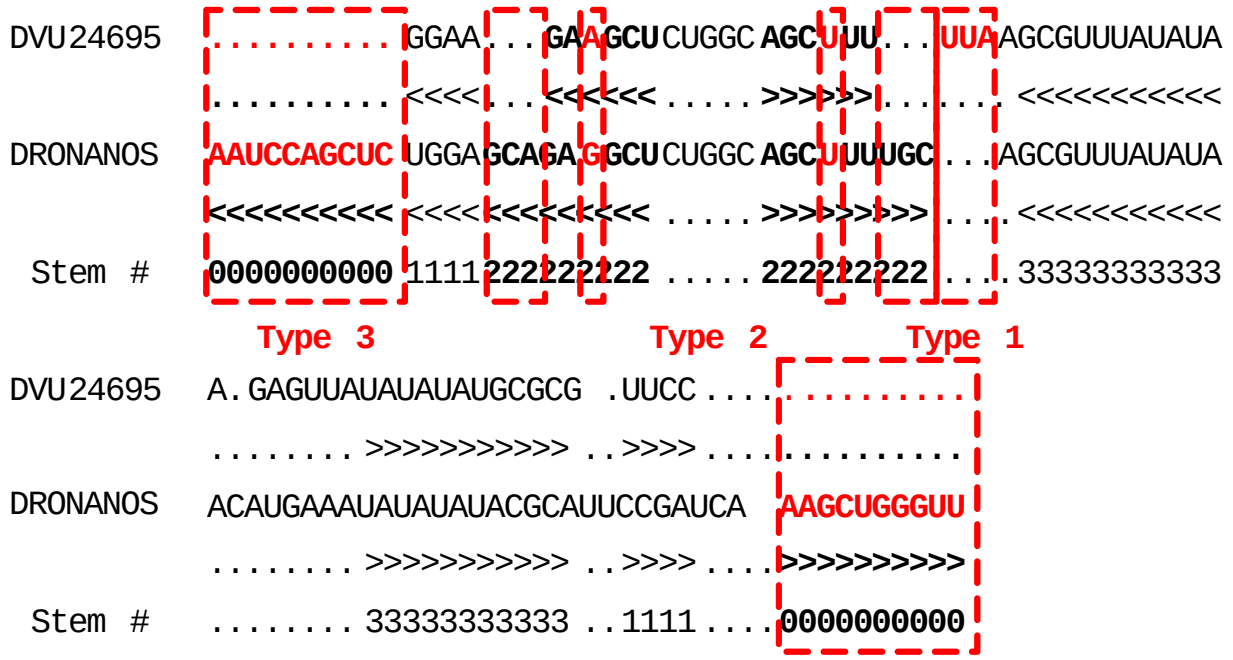


Figure 2.2 Alignment of nanos TCEs from *Drosophila virilis* (DVU24695) and *Drosophila melanogaster* (DRONANOS), copied from [19] and modified for the illustration purpose.

2.5 CONFORMATIONAL GRAPH MODEL

The conformational graph model was originally introduced by Song et al. [20-21] to profile the secondary structure of a family of RNAs including pseudoknots. In this model, the base pair topology of an RNA structure is specified with a mixed graph, with non-directed edges denoting stems and directed edges for loops. In particular, in this graph, each vertex defines either base-pairing region of a stem; two vertices representing two complementary regions (forming a stem) are connected with a nondirected edge. Two vertices defining two regions that are physically next to each other (connected by a loop) are connected with a directed edge (from 5′ to 3′). The individual structural units are stochastically modeled; every stem is associated with a simplified CM and every loop with a profile HMM. The structure graph is capable of modeling RNA structures resulting from multi-body interactions of nucleotides, such as triple helices, as well as pseudoknots. Figure 2.3 shows the structure graph of a typical bacterial tmRNA.

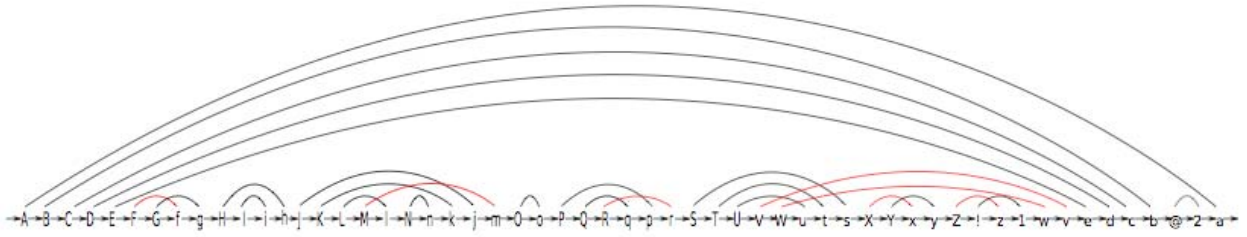


Figure 2.3. The structure graph for the consensus structure of bacterial tmRNAs, which contain four pseudoknots. Each stem is defined by an arc linking the upper and lower cases of the same letter. The red arcs indicate those stems causing pseudoknots.

In this dissertation, two tools, RNATOPS [12] and RNAv [22], were developed based on this conformational graph model. RNATOPS is built directly on this conformational graph model, while RNAv's conformational graph extends this model. In particular, in the extended graph, each vertex represents a contiguous sequence segment, either a loop or one of the two half-stems, making this extended graph model more appropriate for the graph homomorphism alignment. Figure 2.4 shows one example of the conformation graph model used in RNAv.

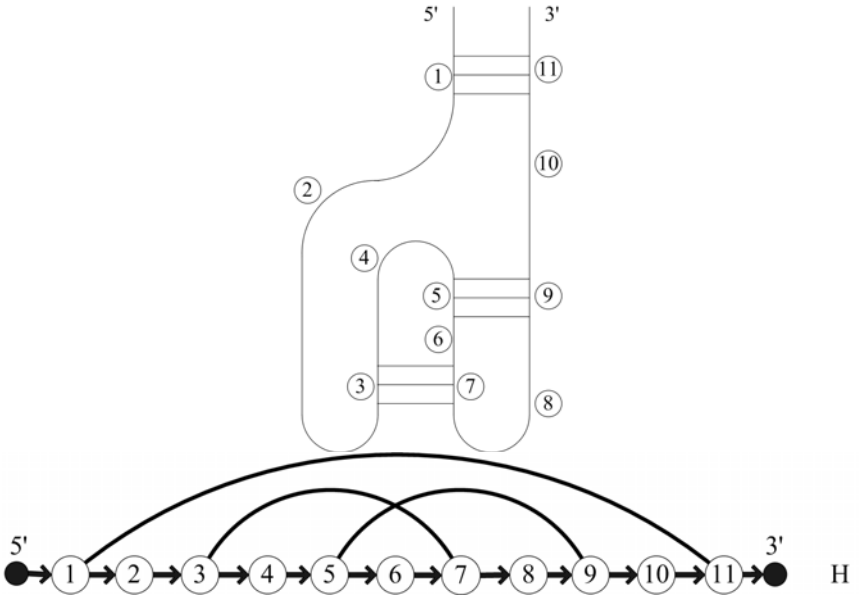


Figure 2.4 A pseudoknot structure and the corresponding conformational graph, H

2.6 SUMMARY OF THE CHAPTER

This chapter introduces some terms for RNA secondary structure, pseudoknot, structural homolog search, RNA structural variation and conformational graph model, which will be used throughout the dissertation.

2.7 REFERENCES

[1] Durbin,R. et al. (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

[2] Meyer, I. M. 2007. A practical guide to the art of RNA gene prediction. Brief. Bioinform. 8:396–414.

[3] Staple DW, Butcher SE. Pseudoknots: RNA structures with diverse functions. PLoS Biol 2005;3:e213

[4] Uemura,Y. et al. 1999. Tree adjoining grammars for RNA structure prediction. Theoret. Comput. Sci., 210, 277–303.

[5] Rivas E, Eddy S, 1999, Pknots A dynamic programming algorithm for RNA structure prediction including pseudoknots. J Mol Biol, 285(5):2053-2068.

[6] Cai, L., Malmberg, R.L., Wu, Y., 2003, Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics 19(Suppl 1):i66-i73.

[7] Hiroshi Matsui, 2005, Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures.

[8] Yuki Kato, et al. 2007, RNA Pseudoknotted Structure Prediction Using Stochastic Multiple Context-Free Grammar.

[9] Brown,M and Wilson,C. (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In Hunter,L. and Klein,T. (eds) Proceedings of Pacific Symposium on Biocomputing. World Scientific Publishing Co, Singapore.

[10] Liu,C. et al. (2006) Efficient annotation of non-coding RNA structures including pseudoknots via automated filters, In Proceedings of Life Science Society Computational Systems Biology Conference (CSB 2006). Imperial College Press, London, pp. 99–110.

[11] Gautheret D., and A. Lambert. 2001. Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. J. Mol. Biol. 313:1003–1011.

[12]. Huang, Z., Y. Wu, J. Robertson, L. Feng, R. Malmberg, and L. Cai. 2008. Fast and accurate search for non-coding rna pseudoknot structures in genomes. Bioinformatics. 24:2281–2287.

[13] PP Gardner, R Giegerich, A comprehensive comparison of comparative RNA structure prediction approaches. BMC Bioinformatics, 5, (2004) 140.

[14] Freyhult, E.K., Bollback, J.P. and Gardner, P.P. (2007) Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA. Genome Res, 17, 117-125.

[15] Griffiths-Jones,S. (2007) Annotating noncoding RNAgenes. Annu. Rev. Genomics Hum. Genet., 8, 279–298.

[16] Lowe TM, Eddy SR. 1997. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res. 25:955–64

[17] Eric P. Nawrocki, Diana L. Kolbe, and Sean R. Eddy, Infernal 1.0: inference of RNA alignments, Bioinformatics. 2009 May 15;25(10):1335-7.

[18] Durbin,R. et al. (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

[19] Holmes I: A probabilistic model for the evolution of RNA structure. BMC Bioinformatics 2004, 5:166.

[20] Song,Y. et al. (2005) Tree decomposition based fast searching for RNA structures with and without pseudoknots. Proc. IEEE Comput. Syst. Bioinform. Conf., IEEE Computer Society Press. 223–234.

[21] Song,Y. et al. (2006) Efficient parameterized algorithms for biopolymer structuresequence alignment. IEEE/ACM Trans. Comput. Biol. Bioinform., 3, 423–431.

[22] Huang,Z., Malmberg,R., Mohebbi, M and Cai,L. 2010. RNAv: Non-coding RNA Secondary Structure Variation Search via Graph Homomorphism, In Proceedings of Computational Systems Bioinformatics Conference (CSB 2010), August, 2010. Vol. 9, p. 56-69.

[23] Srivastava, A., Cai, L., Mrazek, J., and Malmberg, R.L. (2011). Evolution of RNA secondary structure. PLOS One, accepted.

[24] J. Lee and R. Gutell. Diversity of base-pair conformations and their occurrence in rRNA structure and RNA structural motifs. Journal of Molecular Biology. 2004. 344:1225-1249.

[25] N. Leontis, J. Stombaugh, and E. Westhof. The non-watson-crick base pairs and their associated isostericity matrices. Nucleic Acids Research. 2002. 30(16):3497-3531.

[26] Y. Dang, Y. Zhang, and D. Zhang. Statistical parser for RNA secondary structure prediction. In Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005. (6):3399-3403. August 2005. Guangzhou, China.

[27] Bafna,V. and Zhang,S. FastR: fast database search tool for non-coding RNA. In Proceedings of the 3rd IEEE Computational Systems Bioinformatics Conference. Imperial College Press, London, 2004, pp. 52–61.

CHAPTER 3

TREE DECOMPOSITION OF GRAPHS

In the first two chapters, we introduced the conformational graph model and modeled the sequence structure alignment problem as a subgraph isomorphism problem. The solution to the subgraph isomorphism problem is graph tree decomposition and associated dynamic programming algorithms. In this chapter, we will introduce the technique of this solution.

3.1 TREE DECOMPOSITION

Let G=(V, E) be a graph, where V is the vertices set and E is the edges set. A tree decomposition of graph G is a pair $\langle \{X_i \mid i \in I\}, T \rangle$ where each X_i is a subset of V, called a tree bag; T is a tree topology with the elements of X_i as nodes and I is the sets of vertices. It has the following three properties:

1. $\bigcup_{i \in I} X_i = V$;

2. $\forall (u,v) \in E$, $\exists i \in I$ such that $u \in X_i$ and $v \in X_i$;

3. $\forall i, j, k \in I$, if j lies on the path between i and k in T, then $X_i \cap X_k \subseteq X_j$;

The width of $\langle \{X_i \mid i \in I\}, T \rangle$ is defined as $\max_{i \in I} |X_i| - 1$. The treewidth of graph G is the minimum width over all possible tree decompositions of G.

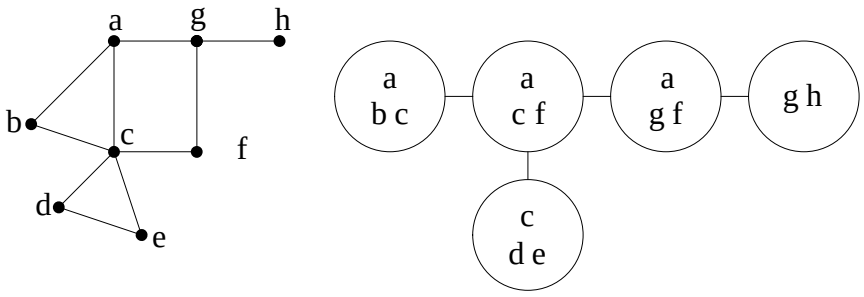


Fig 3.1. An example of a graph, G, with a tree decomposition, T, of treewidth 2 taken from [4].

Tree decomposition was originally introduced by Robertson and Seymour when they studied the graph minor theory [1]. The notion of tree decomposition can be better understood from vertex separator point of view. That is, a valid tree decomposition of a graph is an overlap partition of its vertices into different vertex subsets. Each subset is called a tree bag. Neighboring tree bags may have common vertices and these common vertices can be treated as vertex separators. For example in Fig 3.1, the induced subgraph of {a, b, c} and the induced subgraph of {a, c, f} are separated by vertex separator set {a, c}. Also induced subgraph of {a, b, c} and the induced subgraph of {c, d, e} are separated by vertex separator set {c}.

Tree decomposition provides a different view of graph topology, mapping a graph into a tree representation. This approach can be used to develop efficient algorithms to solve many NP-hard problems on graphs of small tree-width.

In [11] it shows that for almost all RNA secondary structures (including pseudoknots), the yielded tree decompositions of the conformational graph have small tree width $t, t \leq 4$. So efficient algorithms can be developed to solve the optimization problem based on this conformational graph.

3.2 TREE DECOMPOSITION AND ITS APPLICATION

Generally, a tree decomposition based algorithm has the following two procedures [4]: (1) Find a tree decomposition for the input graph (with tree-width bounded by a constant, not necessarily the optimal). (2) A dynamic programming algorithm is executed on this tree decomposition.

The general idea of a tree decomposition based dynamic programming algorithm has been explained in [4, 5, 6]. Here we will use one example, the MAXIMUM INDEPENDENT SET problem, to explain how to apply the tree decomposition based dynamic programming algorithm to solve it. We redraw the tree decomposition T of the graph G (Fig. 3.2). Later we will discuss two issues related with the application of tree decomposition: how to get a good tree decomposition and how to solve a memory issue in the dynamic programming procedure.

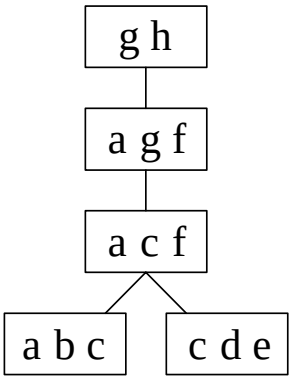


Fig. 3.2. Tree decomposition, T, of the graph, G, in Fig. 3.1.

3.2.1 MAXIMUM INDEPENDENT SET

An independent set in a graph is a set of vertices such that any two vertices in this set are not connected in the graph. The MAXIMUM INDEPENDENT SET problem is to find such independent set with the maximum size. This problem is NP-hard [12], however, it can be solved in linear time on graphs which has treewidth bounded by a constant.

We assume a graph G=(V, E) has a tree decomposition, T, with tree width of t, and T is a binary rooted tree, without loss of generality (later we will explain why this assumption will not affect the time/space analysis result). For each tree bag, X_i, of T, we compute a dynamic

programming table. Values assigned to each vertex in this tree bag depend on the optimization problem to be solved. In the MAXIMUM INDEPENDENT SET problem, a value of the vertex indicates either this vertex will be chosen into the independent set (e.g. of the value 1) or not (e.g. of the value 0). The size of this table is a function of the number of vertices in this tree bag and the number of values a vertex can be assigned. If t is the treewidth of this graph, then the upper bound of the table size will be 2^{t+1}. Each entry of the table corresponds to a subset of vertices and for each entry of the table, two properties will be determined. One is whether the current subset of vertices corresponds to a valid independent set. The other is about the size of the independent set, formed by a subset of vertices in the tree bag.

We compute the dynamic programming table in the bottom-up fashion. This is a post-order traversal of tree decomposition T. The table for tree bag X_i is computed after the tables of all descendants of X_i have been computed. In particular, for a leaf tree bag, the dynamic programming algorithm enumerates all possible combinations of vertices in the bag as table entries and determines the values of *valid* and *size* for each entry. To determine the value for *valid*, the algorithm needs to check whether the vertices picked in the current entry form an independent set in the graph G. If it is valid, the algorithm determines the value of *size* for the number of the independent set. For example, to compute the dynamic programming table for the tree bag {a, b, c} in Fig. 3.3, there are 3 vertices in this tree bag and totally 8 ($=2^3$) entries in this table. For the 4^{th} row (a=0, b=c=1), because in graph G, b and c are connected in the graph; hence, this combination is invalid. The value of *valid* is set to be X. For the 5^{th} row (a=1, b=c=0) represents a valid independent set, therefore *valid*= √ and also the size of this independent set is 1.

For a non-leaf tree bag X_i, the scenario becomes a little more complicated because the algorithm needs to query the valid entries from its child tree bags. Without loss of generality, we assume this non-leaf tree bag X_i, has two children, tree bag X_j, and tree bag X_k. Three computations need to be done to determine the value of *valid* and *size* for each entry in the table of tree bag X_i. First, the algorithm determines the value of *valid* (V_i) and *size* (S_i) for each entry in the tree bag X_i following the same idea of the leaf tree bag. Second, the algorithm determines the value of *valid* (V_j) and *size* (S_j) by querying X_j based on the value for those common vertices in $X_i \cap X_j$. In particular, the value of S_j will be the max value of size by querying the dynamic programming table of X_j using the values from X_i for those common vertices in $X_i \cap X_j$. We need to avoid double counting those separator vertices into the size of the overall independent set. If such S_j can be found, the algorithm sets the value of *valid* (V_i) to be $\surd$, otherwise it is $\times$. Third, similarly the algorithm determines the value of *valid* (V_k) querying X_k based on the value for those common vertices in $X_i \cap X_k$. Then the value of *size*, S_i' will be $S_i+S_j+S_k$ and the value of *valid*, V_i' will be V_i AND V_j AND V_k.

We take the computation of *valid* and *size* for tree bag {a, c, f} as an example to illustrate it. For the entry of picking f and not picking a and c (a=0, c=0 and f=1), the algorithm checks the graph and determines this combination is a valid independent set. Because $X_i \cap X_j$={a, c}, the algorithm picks the entry of E_j (a=0, b=1, c=0) from X_j; also because $X_i \cap X_k$={c}, the algorithm picks the entry of E_k (c=0, d=1, e=0) from X_k. Finally the algorithm computes the value of *valid* of the current entry E_i to be $\surd$ and sets *size* to be 3.

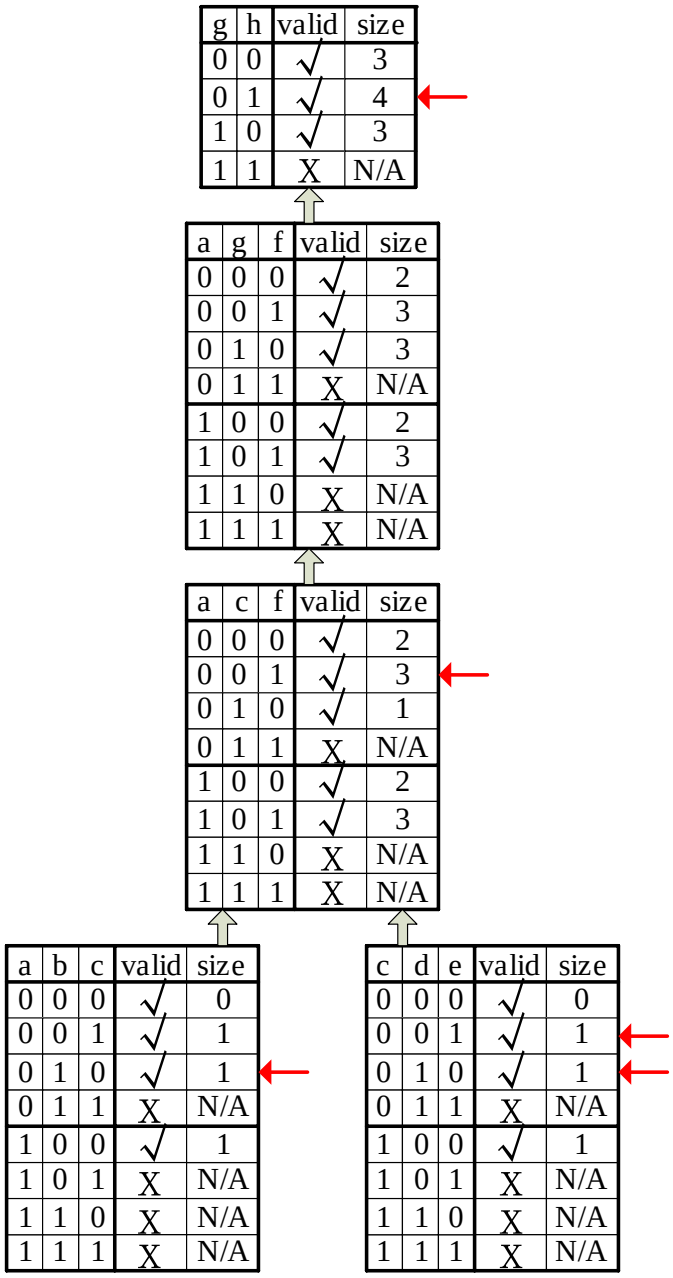


Fig. 3.3. Bottom-up tree decomposition based dynamic programming

The optimal result will be accessed from the root tree bag. The algorithm will query the table and pick the valid combination with the maximum *size* value. Later, a top-down trace back procedure can be executed to find all the vertices in this maximum independent set. In this example (Fig 3.3), the optimal result can be accessed from the 2nd row (g=0, h=1) and the MIS

optimal result can be (a=0,b=1,c=0,d=0,e=1,f=1,g=0,h=1), which means the vertex set is {b,e,f,h}. Based on the previous analysis, it is not difficult to see that the time complexity of this dynamic programming algorithm in a graph G will be bounded by $O(2^{t+1}*n)$ where n is the number of tree bags in T, and the memory assumption is also bounded by $O(2^{t+1}*n)$. Because any non-redundant tree decomposition of a graph with n vertices has at most n tree nodes [7], and also a binary tree decomposition can be constructed in polynomial time, the running time for a binary tree decomposition will be bounded by $O(2^{t+1}*|V|)$ and memory will be bounded by $O(2^{t+1}*|V|)$ for a graph G=(V, E).

3.3 DISCUSSION

From the above analyses, it is not hard to see that the running time of this dynamic programming is exponential in the tree-width of the graph. Moreover, the memory space consumption (the dynamic programming tables) is also exponential in the tree-width. These are the two factors that need to be considered to make the tree decomposition based dynamic programming algorithm efficient.

Indeed, the main bottleneck for the efficiency often is memory consumption but not running time [16]. So here we will focus on the discussion about how to save memory in the dynamic programming algorithm. For those who are interested in getting the small tree-width, please refer to Bodlaender and Koster's paper [2, 3].

3.3.1 HOW TO SOLVE THE MEMORY LIMITATION PROBLEM

Aspvall et al. [8] dealt with this space problem by an optimal traversal of the decomposition tree in order to minimize the number of dynamic programming tables stored simultaneously, but this technique seems to work in the decision problem only [9]. Nadja Betzler [9] proposed the anchor technique to save the memory in dynamic programming; they tested on the nice tree

decompositions and reported the 60% memory savings. We believe that nice tree decomposition is good for theoretical analysis, but not ideal to be applied in the dynamic programming phase. This is because one nice tree decomposition has many tree nodes redundancy. In the following we provides some mapping functions to save the memory on the general tree decomposition [10].

Our solution to save the memory when doing tree decomposition based dynamic programming was the main idea as follows. Vertices in each tree bag will be classified into two categories. One is the set of vertices that occur in both this and its parent tree bags (overlap part). The other is the rest of the vertices (non-overlap part). During the dynamic programming, the computation information in each dynamic programming table is stored for the following trace-back phase. Memory for the non-overlap vertex set can be represented by one column in the table to store the index for the optimal combination of the non-overlap vertices because the optimal combination will only be computed once during the bottom-up process. While in the trace-back process, the index for the non-overlap vertices will be used to compute the optimal combination for those non-overlap vertices. This memory saving strategy has been applied in the program of RNATOPS [11] we developed in chapter 4. The other memory saving strategy is: for those overlap vertices in the tree bag, space is only allocated for three properties: *valid*, *size* and *the pointer to the max value of non-overlap combination*, space for the combination of the overlap vertices can be saved because this combination can be computed by the index of the overlap vertices in the table. This allows the space consumption for each internal tree bag to be reduced from the original $2^{t+1}*2$ to the current $2^{s}*3$ where t is the tree-width and s is the number of vertex shared with its parent tree bag, $s \le t$. In total, the space consumption will be reduced from $2^{t+1}*2*|V|$ to $2^{s}*3*|V|_{Internal\_Tree\_Bag} + 2^{t+1}*2*|V|_{Leaf\_Tree\_Bag}$.

3.4 SUMMARY OF THE CHAPTER

This chapter reviews the theoretical techniques that will be used in the following chapters to solve the RNA pseudoknot and RNA secondary structure variation search problem. They are the graph treewidth and its dynamic programming algorithm. The example of the MAXIMUM INDEPENDENT SET problem is presented to illustrate the tree decomposition based dynamic programming algorithm. The issue arising from the application, space limitation, is discussed.

3.5 REFERENCES

[1] N. Robertson and P.D.Seymour, Graph Minors II. Algorithmic aspects of tree-width. J. Algorithms, 7:309 322, 1986.

[2] Hans L. Bodlaender, Arie M. C. A. Koster. Treewidth computations I. Upper bounds. Inf. Comput., 2010: 259~275.

[3] Hans L. Bodlaender, Arie M. C. A. Koster. Treewidth computations II. Lower bounds. techreport, 2010. http://www.cs.uu.nl/research/techreps/repo/CS-2010/2010-022.pdf.

[4] H.L. Bodlaender and A. Koster, Combinatorial optimization of graphs of bounded treewidth,. The Computer Journal (2007), 631–643.

[5] P Heggernes, Treewidth, partial k-trees, and chordal graphs, September 26, 2006. http://www.ii.uib.no/~pinar/chordal.pdf.

[6] Niedermeier, R. Invitation to Fixed-Parameter Algorithms. Oxford University Press, 2006

[7] J. Kleinberg and E. Tardos. Algorithm Design. Addison-Wesley, 2005.

[8] B. Aspvall, A. Proskurowski, and J. A. Telle. Memory requirements for table computations in partial k-tree algorithms. Algorithmica 27(3): 382–394, 2000.

[9] Nadja Betzler, Rolf Niedermeier, Johannes Uhlmann: Tree decompositions of graphs: Saving memory in dynamic programming. Discrete Optimization 3(3): 220-229 (2006).

[10] Qi Li, Graph Tree Decomposition Enabled Biopolymer Folding, master thesis, UGA, 2010.

[11] Huang, Z., Y. Wu, J. Robertson, L. Feng, R. Malmberg, and L. Cai. 2008. Fast and accurate search for non-coding rna pseudoknot structures in genomes. Bioinformatics. 24:2281–2287.

[12] M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman and co., New York, 1979.

CHAPTER 4

FAST AND ACCURATE SEARCH FOR NON-CODING RNA PSEUDOKNOT STRUCTURES IN GENOMES[1]

[1]Zhibin Huang, Yong Wu, Joseph Robertson, Liang Feng, Russell L. Malmberg and Liming Cai, 2008, Bioinformatics. 24:2281–2287.
Reprinted here with permission of publisher.

4.1 ABSTRACT

Motivation: Searching genomes for non-coding RNAs (ncRNAs) by their secondary structure has become an important goal for bioinformatics. For pseudoknot-free structures, ncRNA search can be effective based on the covariance model and CYK-type dynamic programming. However, the computational difficulty in aligning an RNA sequence to a pseudoknot has prohibited fast and accurate search of arbitrary RNA structures. Our previous work introduced a graph model for RNA pseudoknots and proposed to solve the structure–sequence alignment by graph optimization. Given k candidate regions in the target sequence for each of the n stems in the structure, we could compute a best alignment in time $O(k^{t+1}n)$ based upon a tree width t decomposition of the structure graph. However, to implement this method to programs that can routinely perform fast yet accurate RNA pseudoknot searches, we need novel heuristics to ensure that, without degrading the accuracy, only a small number of stem candidates need to be examined and a tree decomposition of a small tree width can always be found for the structure graph.

Results: The current work builds on the previous one with newly developed preprocessing algorithms to reduce the values for parameters k and t and to implement the search method into a practical program, called RNATOPS, for RNA pseudoknot search. In particular, we introduce techniques, based on probabilistic profiling and distance penalty functions, which can identify for every stem just a small number k (e.g. $k \leq 10$) of plausible regions in the target sequence to which the stem needs to align. We also devised a specialized tree decomposition algorithm that can yield tree decomposition of small tree width t (e.g. $t \leq 4$) for almost all RNA structure graphs. Our experiments show that with RNATOPS it is possible to routinely search prokaryotic

and eukaryotic genomes for specific RNA structures of medium to large sizes, including pseudoknots, with high sensitivity and high specificity, and in a reasonable amount of time. Availability: The source code in C++ for RNATOPS is available at www.uga.edu/RNA-Informatics/software/rnatops/.

4.2 INTRODUCTION

Non-coding RNAs (ncRNAs) have been shown to be involved in many biological processes including gene regulation, chromosome replication and RNA modification (Frank and Pace, 1998; Nguyen et al., 2001; Yang et al., 2001). Searching genomes using computational methods has become important for annotation of ncRNAs (Griffiths-Jones, 2007; Hofacker, 2006; Lowe and Eddy, 1997; Rivas and Eddy, 2001; Rivas et al., 2001; Washietl et al., 2005). In general, to annotate an individual genome for a specific family of ncRNAs, a computational tool needs to scan through the genome and align its sequence segments to some structure model for the ncRNA family. Those segments with significant alignment scores are then reported as the results. An algorithm that can perform an accurate sequence–structure alignment is thus the core of such a searching tool.

A few programs (Brown and Wilson, 1996; Klein and Eddy, 2003; Liu et al., 2006; Lowe and Eddy,1997) have been developed for genome annotation using the covariance model (CM) introduced by Eddy and Durbin (1994). Based on a CM, the optimal alignment between a sequence and a pseudoknot-free structure can be performed with a dynamic programming algorithm in $O(WN^3)$, where W is the size of the model and N is the length of the sequence. In particular, RSEARCH (Klein and Eddy, 2003) and Infernal (http://infernal.janelia.org/) are two programs that can perform such searches. CM-based methods can achieve high searching accuracy; however, due to the time complexity needed for sequence–structure alignment, a CM-

based search may be inefficient on complex or large RNAstructures. Further, pseudoknot structures, which contain at least two interweaving stems, cannot be modeled with CMs. Searches on genomes can be speeded up with filtering methods (Bafna and Zhang, 2004; Lowe and Eddy, 1997; Weinberg and Ruzzo, 2004, 2006; Zhang et al., 2005). Sometimes it is possible to efficiently remove genome segments unlikely to contain the desired pattern. For example, in tRNAscan-SE (Lowe and Eddy, 1997), two efficient filters are used to preprocess a genome and remove the part that is unlikely to contain the searched tRNA structure; the remaining part of the genome is then scanned with a CM to identify the tRNA. FastR (Bafna and Zhang, 2004) considers the structural units of an RNA structure; it evaluates the specificity of each structural unit and construct filters based on the specificity of these structural units. In Weinberg and Ruzzo (2004), an algorithm is developed to safely break the base pairs in an RNA structure and automatically select filters from the resulting Hidden Markov Model (HMM). These approaches have significantly improved the computational efficiency of genome searches.

RNA structures that contain pseudoknots pose special problems. A number of creative approaches (Cai et al., 2003; Rivas and Eddy, 1999, 2000; Uemura et al., 1999) have been tried to model the crossing stems of pseudoknots; however, the time and space complexities for optimal sequence–structure alignment based on these models are $O(N^4)$ or $O(N^5)$. These models are not practical for efficient searching. Intersecting CMs have been proposed for pseudoknots (Brown and Wilson, 1996), and used to search small genomes (Liu et al., 2006), but these have the same efficiency problem. Several heuristic search methods have been developed that can work with RNAs containing pseudoknots; as heuristics, each has some limitations. For example, ERPIN (Gautheret and Lambert, 2001), considers the stem loops contained in a secondary structure. The genome is then scanned to find the possible hit locations for each stem

loop.Ahit for the overall structure is reported when there exists a combination of hit locations for different stem loops that conform with the overall structure. However, ERPIN does not allow gaps in the alignment and thus may have low sensitivity when the target is a remote homolog of the query structure model.

Our previous work introduced a graph modeling method that can profile the secondary structure of a family of RNAs including pseudoknots (Song et al., 2005, 2006). In this method, the topology of an RNA structure is specified with a mixed graph, with non-directed edges denoting stems and directed edges for loops. With this model, we proposed to efficiently solve the structure–sequence alignment problem, including pseudoknots, by exploiting the small tree width (Robertson and Seymour, 1986) demonstrated by the structure graphs of almost all existing RNA pseudoknots. Theoretically, given k (pairs of) regions as candidates for each of the n stems in the structure and given a tree decomposition of tree width t for the structure graph, the alignment can be computed in time $O(k^{t+1}n)$. However, to implement the algorithm into computer programs that can routinely perform fast, accurate RNApseudoknot search, heuristics for the preprocessing steps need to be able to associate results with small values of parameters k and t while maximizing search accuracy.

In this article, we present our current work, built upon the previous one, to develop a practical program, called RNATOPS, for RNA pseudoknot search. In this work, we have introduced new, effective heuristic techniques for generating stem candidates and for tree decomposition of RNA structure graphs. In particular, parameter k can be chosen relatively small (e.g. $k \leq 10$) to ensure both accuracy and efficiency of the search. The alignment algorithm (and thus the search algorithm) runs in time $O(k^{t+1}n)$, linear in the number n of stems in the profiled RNA structure. It is scalable with the complexity of the profiled structure because the

yielded tree decompositions have small tree width t, $t \leq 4$, for almost all RNA secondary structures (including pseudoknots). In this article, we evaluate RNATOPS with search tests conducted on several medium to large size RNAs (including pseudoknots) and make comparisons with existing RNA structure search programs such as Infernal.

4.3 APPROACH

We refer the reader to the publications (Song et al., 2005, 2006) for detailed discussions of our graph modeling method for RNA structures and on the solution to structure–sequence alignment based on tree decomposition of the structure graph. In this section, we give a brief recap of the necessary notions and techniques relevant to the current article.We then present the new heuristic techniques for stem candidate identification and for tree decomposition designated for RNAstructure graphs. These heuristic techniques aim at achieving a fast structure–sequence alignment without degrading the accuracy.

4.3.1 A GRAPH MODEL FOR STRUCTURE SEARCH

Our structure model based on a mixed graph specifies the consensus structure of an RNA family as a relation among all involved structural units: stems and loops. In this graph, each vertex defines either base pairing regions of a stem; two vertices representing two complementary regions (forming a stem) are connected with a nondirected edge. Two vertices defining two regions that are physically next to each other (forming a loop) are connected with a directed edge (from $5'$ to $3'$). The individual structural units are stochastically modeled; every stem is associated with a simplified CM and every loop with a profile HMM. The structure graph is capable of modeling RNA structures resulting from multi-body interactions of nucleotides, such as triple helices, as well as pseudoknots. Figure 4.1 in Supplementary Material shows the structure graph of a typical bacterial tmRNA.

Searching in a target genome consists of sliding a window of appropriate size along the target genome, then testing for a possible alignment of the structural model with the sequence segment within the current window. With the graph model, the structure–sequence alignment is identical to the task of finding the optimal subgraph of a graph G isomorphic to another graph H, where H is the RNA structure graph and G is constructed from the target sequence in a preprocessing step. We proposed two methods to cope with the computational intractability of the subgraph isomorphic problem. One method was to pre-identify in the target sequence top k candidates for every stem in the structure. The other method was to tree decompose the structure graph. Based upon a tree decomposition, a dynamic programming algorithm could solve the subgraph isomorphic (thus the structure–sequence) problem in theoretical time $O(k^{t+1}n)$, where n is the number of stems in the structure and t is the tree width of the graph tree decomposition (Song et al., 2005, 2006). This article presents new heuristic techniques to support these two methods.

4.3.2 MODEL TRAINING

Model training involves defining the structure graph, individual CMs and profile HMMs from a set of training RNA sequences given in a pasta file. The pasta format (pairing plus fasta) is a representation we developed for multiple structural alignment and consensus structure of RNA sequences (Fig. 4.2 in Supplementary Material). It labels stem positions with an upper case letter for one side, the corresponding lower case letter for the other side. The first line of the file denotes the consensus structure using matching (upper and lower case) letters for conserved base pairs and '.'s for unpaired nucleotides or possibly consensus insertions. Representation with pairing letters has the advantage of being able to denote arbitrary RNA structures, including pseudoknots and triple helices.A structure graph is produced from the consensus structure, where

one vertex is for one letter, one non-directed edge connects the two vertices of matched letters and one directed edge connects two neighboring letters (from $5'$ to $3'$, Fig. 4.1 in Supplementary Material).

The rest of the lines in the pasta file are RNA sequences structurally aligned to the consensus structure, possibly containing '-'s for deletions. Individual CMs and profiles HMMs are constructed from the multiple structure alignment as follows. Every stem of base-paired regions (with matching letters) produces one simplified CM that does not contain bifurcation rules or rules for the sequence connecting the two base-paired regions. One profileHMM is generated from every two neighboring base regions. The profile HMM allows possible match, insertion and deletion states in every column of the multiple alignment. The parameters of these stochastic models are computed from the multiple structural alignment using the maximum likelihood method. To avoid over-fitting the models, we incorporate background statistics. In particular, we allow pseudocounts for nucleotides in the match, insertion and deletion states of the profile HMM. For the simplified CM, a 4×4 prior probability matrix P_p for base pairs and a weighting parameter w are introduced so that the probability of a base pair $P(x, y)$ is defined as the weighted sum $wP_t(x, y) + (1-w)P_p(x, y)$, where P_t is the base pair probability matrix obtained from the training data.

4.3.3 IDENTIFYING STEM CANDIDATES

The sequence segment within the sliding window is preprocessed to identify top k candidates for the CM of every stem. Given a CM modeling some consensus stem, the score of every possible structural motif within the window aligned to the model is computed (Fig. 4.3 in Supplementary Material). Candidates can be found by a simple dynamic programming algorithm; we describe here four heuristic techniques developed to ensure that the correct motif structure for

the CM, if it does exist in the sequence, is highly likely to be among the selected top k candidates for some small value of k.

(1) Regions from which candidates can be selected are constrained according to the statistical distribution of the consensus stem in the sample (training sequences). In particular, we assume a Gaussian distribution for the position of the consensus stem in the RNA structure. The constrained region for the correct motif of the consensus stem is within a certain number (e.g. 3) of the SD of the average position.

(2) For training sequences that demonstrate a large SD for the position of some consensus stem, training sequences are partitioned into clusters, each with a small SD for the stem position. Therefore, more than one (constrained) region may be derived for the correct motif of the consensus stem.

(3) The candidates so identified are then ranked again according to statistical distributions of various length parameters associated with a consensus stem, including the length of the stem, the distance between the two stem arms and the head and tail offsets. The scores of every possible motif candidate c of the CM M are recalculated according to the formula: $S(c,M) = uA(c,M) + (1-u)P(c,M)$, where $A(c,M)$ is the logodds score from the alignment, $P(c,M)$ is the penalty function for the deviations of all lengths list above from their means and u, $0 \le u \le 1$, is a weighting parameter. In particular, $P(c,M)$ is computed based on the log score $\log(1/cK^2)$, where $K = |l-\mu|/\sigma \ge 1$ for the length l deviating from mean μ (with a SD σ) and c is a selected constant.

(4) Finally, since it is possible that several structural motifs, heavily overlapping in their positions, may all have decent alignment scores with respect to a stem model, it suffices to

record only one representative for them. Strategies have been used to select representatives and to ensure a low value for k, the number of top candidates.

4.3.4 TREE DECOMPOSITION FOR STRUCTURE GRAPHS

With our model, almost all ncRNAs have structure graphs of small tree width. However, finding the optimal tree decomposition (one with the smallest tree width) is NP-hard. Available efficient tree decomposition algorithms are for general graphs and usually do not guarantee the optimal tree width. For RNA structure graphs, we develop a linear-time greedy algorithm that can yield tree decomposition of tree width almost always bounded by 4. An earlier version of this algorithm was given in (Song et al., 2005), but it used the idea of minimum fill-in and may produce decompositions of unnecessarily larger tree widths. We present a self-contained version of the algorithm here.

First, the algorithm removes arcs (i.e. non-directed edges) in the structure graph that cross with other arcs. It does this by greedily removing the arc crossing the most other arcs and repeating the step on the remaining graph until there is no crossing arc in the graph (Fig. 4.4a and b in Supplementary Material). This step actually removes stems involved in pseudoknots in the corresponding RNA structure; a crossing arc-free structure graph corresponds to a pseudoknot-free RNA structure. Such a graph is an outer-planar graph that has tree width 2, whose optimal tree decomposition can be found as follows.

Note that in a structure graph, the vertices are arranged in the direction of from $5'$ to $3'$ (left to right in the figures) based on the directed edge relation. We also add the source s and sink t as the left most and the right most vertices, respectively. We use notation H_b^a to represent the subgraph induced by the set of vertices 'from' vertex a 'to' vertex b (inclusive, from $5'$ to $3'$).

Then to decompose the subgraph H_t^s, the algorithm handles the following three major scenarios recursively (and the recursive process terminates when the considered subgraph is empty).

(1) If (s,X) is a directed edge but (x,t) is not, where (X,x) is an arc (Fig. 4.5a in Supplementary Material), then the root node {s,t} has child node {s,x,t}, which in turn has child node {s,X,x} (Fig. 4.5b in Supplementary Material). Node {s,X,x} will be the root for the subtree generated from subgraph H_x^X and node {s,x,t} will be the root for the subtree generated from subgraph H_t^x.

(2) If (s,X) and (x,t) both are directed edge, where (X,x) is an arc (Fig. 4.5c in Supplementary Material), then the root {s,t} has child node {s,X,t}, which in turn has child node {X,x,t}. Node {X,x,t} will be the root for the subtree generated from subgraph H_x^X (Fig. 4.5d in Supplementary Material).

(3) If (s,X) is a directed edge but (X,x) is not an arc (Fig. 4.5e in Supplementary Material), then the root {s,t} has a child node {s,X,t}, which in turn will be the root for the subtree generated from subgraph H_t^X (Fig. 4.5f in Supplementary Material).

The algorithm modifies the resulting tree decomposition as follows. For every removed arc (v, v'), the algorithm identifies two nodes, one containing vertex v and another containing its counterpart v'. For every tree node on the path from the former node to the latter, the algorithm adds v to it (Fig. 4.6 in Supplementary Material). This gives a tree decomposition for the original structure graph.

4.4 IMPLEMENTATION

RNATOPS, implemented in language C++, has been compiled and tested on several systems, including Desktop Linux computers, a Linux cluster and a SUN workstation running SunOS 5.1.

4.5 EVALUATION

To evaluate the search program and the effective of the heuristics, we tested RNATOPS using four types of RNAs of medium to large sizes: bacterial tmRNA, bacterial RNaseP type B RNA, yeast telomerase RNA and bacterial 16S rRNA. We compare both search accuracy and efficiency of RNATOPS with those of Infernal and FastR, two of the best known general-purpose programs for RNAstructure search.

4.5.1 DATA PREPARATION AND TESTS CONDUCTED

Bacterial tmRNAs (Moore and Sauer, 2007; Nameki et al., 1999) have a complex structure containing four pseudoknots; there are 178 molecules in the Rfam (Griffiths-Jones et al., 2005) seed alignment with an average length of 364 bases (Fig. 4.1 in Supplementary Material). The tmRNA sequences have variations in structure with certain stem loops present in some sequences and absent in others. We extracted a subset of 43 tmRNA sequences from the 178 molecules in the alignment, which did not differ from each other in the presence or absence of any stem loops, and for which the entire bacterial genome sequence was available; columns consisting entirely of gaps were then removed from the alignment.

RNaseP, bacterial type B, RNAs have multiple stem loops and one sophisticated pseudoknot (Brown, 1999; Harris et al., 2001; Fig. 4.7 in Supplementary Material). There are 31 sequences of average length 367 in the Rfam seed alignment.We extracted a subset of 10 sequences which did not differ from each other in the presence or absence of any stem loops; the full genome sequence was available for 7 of the 10.

Yeast telomerase RNAs contain a conserved, essential, pseudoknot within a large stem loop (Chen and Greider, 2004). We used an alignment, of length 834, for this region (Dandjinou et al., 2004) of six Saccharomyces species telomerase RNAs. While the genome of S.cerevisiae has

been completely sequenced, those of the other Saccharomyces species are available in varying degrees of completeness and assembly.We were able to collect four Saccharomyces genomes total, three in addition to S. cerevisiae, to search.

The bacterial 16S rRNA is a conserved molecule which has been extensively used for phylogenetic studies of bacteria. We obtained an alignment (of 1570 bp) of the 16S rRNA for gammaproteobacteria from the ribosomal database (Cole et al., 2007); from this we selected those sequences which contained an identical match in a fully sequenced bacterial genome. Although many gammaproteobacteria genomes have been sequenced, for only 12 was there an exact match between the database sequence and a genomic sequence, which we required to take advantage of the expert alignment from the database. These sequences were used as the training set.

For all the genomic searches, we followed a cross-validation approach in which the RNA found in a genomic sequence was removed from the alignment, and the remaining sequences were used as a training set for a search on that genome.

To search genomes of a considerable length, we identified highly conserved motifs of the RNAmolecules, then searched the genomes with these, after which we examined the region around a potential hit for a structural match to the whole molecule. We note that a program that can automatically identify a conserved motif as the optimal filter is currently being developed for RNATOPS.

4.5.2 COMPARISON TO OTHER SEARCH PROGRAMS

To compare with Infernal (infernal.janelia.org), we downloaded Infernal from its website, compiled it and installed it, and compared its performance on one of the same Linux computers

we used for testing of RNATOPS. Both Infernal and RNATOPS use multiple structural alignments for model training and use filters to speed up the search.

We used FastR (Bafna and Zhang, 2004; Zhang et al., 2005) through job submission at a website. As such, it is difficult for us to compare the performance of FastR on a server of unknown configuration and numbers of cpus with the performance of RNATOPS. During the times we tested it, our analyses were the only ones listed in the job queue. We estimated the time of the run from the time of submission and the time at which the job finished e-mail was sent. The user can pick from pre-defined profiles for searching. It is unknown to us if these profiles included the tmRNAs for the genomes we tested. Hence these FastR tests may or may not correspond to the training sets we used, in which we left out the RNAs for the genome targeted for searching.

4.5.3 SEARCH ACCURACY

4.5.3.1 BACTERIAL TMRNAS

We searched 43 bacterial genomes with RNATOPS for tmRNAs using a leave-one-out cross-validation approach. Table 4.1 in Supplementary Material gives a comparison of the results achieved with RNATOPS with those of Infernal. RNATOPS was evaluated with varying parameter k, the number of candidate regions examined for each stem in the structure. Increasing k from 10 to 15 to 25 increased the sensitivity of the whole structure search, but also increased the time taken. For example, at $k = 10$, the bacterial genome searches gained 88% sensitivity and 100% specificity; at $k = 25$, the sensitivity increased to 98%. Infernal had 100% sensitivity and specificity for these searches with comparable times spent.

We observed that the tmRNAs missed by RNATOPS at the low k values generally had one or more portions in the structure, which significantly deviated from the consensus structure.

In particular, several stems in these sequences consisted of mainly rare, non-canonical base pairs, which may have been placed in pairing positions during the multiple alignment process.

We also compared the alignments of the tmRNA structures found by Infernal and RNATOPS. Four structures identified by RNATOPS have stem alignments off their correct positions for more than a few nucleotides in their alignments; Infernal identified seven such structures. There are in total nine such stem misalignments in the structures identified by RNATOPS; there were total 17 in those structures identified by Infernal. In addition, because Infernal is based on the pseudonot-free CM, in a structure alignment, regions 'belonging to' a pseudoknot may be mistakenly aligned to pseudoknot-free substructures. In particular, in this set of search tests, there were totally five such incorrect assignments found in the search results of Infernal while the issue was not raised on RNATOPS (Table 4.6 in Supplementary Material).

We also tried the search with FastR web server, which includes tmRNAs as a profile. We selected one bacterial genome on which RNATOPS successfully found the tmRNA, and one genome on which RNATOPS failed to find the tmRNA, then submitted these to the FastR server. FastR gave the same results as RNATOPS with these two sequences, finding the structure in one sequence and missing it in the other (Table 4.2 in Supplementary Material), again suggesting there is something unusual about the tmRNA that both programs missed. Several additional bacterial genomes were submitted to the FastR server, but no results were returned.

4.5.3.2 BACTERIAL RNASEP (BACT. B) RNAS

The bacterial RnaseP (Bact. B) RNA is similar in size to the tmRNAs, but has a more complex pseudoknot structure. Both the RNATOPS and Infernal programs had 100% sensitivity and 100% specificity in finding the RNaseP RNAs in the seven genomes tested (Table 4.3 in Supplementary Material); RNATOPS identified two structures whose alignments put in total four

stems off their correct positions by more than a few nucleotides (Table 4.6 in Supplementary Material).A comparison with the tmRNA results suggests that the more complex pseudoknot structure in RNaseP (Bact. B) was handled well by RNATOPS, with less than a doubling in time taken for similar sized genomes, while Infernal took about nine times as long.

4.5.3.3 SACCHAROMYCES TELOMERASE RNAS

The conserved core region of Saccharomyces telomerase RNAs is more than twice as long as the bacterial tmRNAs or RNaseP RNAs, and the Saccharomyces genomes are 2 to 10 times larger than the bacterial genomes tested. The pseudoknot structure itself is not complex, but it is contained within a stem-loop and some additional stem-loops are present. Both programs found the four Saccharomyces fungal telomerase RNAs perfectly in their genomes; RNATOPS took from 5.5 to 6.4 min, while Infernal took from 295 to 654 min for the same searches (Table 4.4 in Supplementary Material).

4.5.3.4 BACTERIAL 16S RRNAS

The bacterial 16S rRNAs are the longest molecule we tested with lengths around 1500 bp. The results were similar to the telomerase and RNAseP RNAs, with both RNATOPS and Infernal finding the target with perfect specificity and sensitivity, but with RNATOPS performing the search in an average of 14.1 min as opposed to 88 min for Infernal (Table 4.5 in Supplementary Material).

4.5.4 EFFICIENCY

The theoretical time of the search method can be expressed as $O(T_a N)$, where T_a is the time needed for the structure alignment between the structure model and the sequence segment within the window sliding through the genome of N nucleotides. T_a actually consists of two parts: the time for the preprocessing step and the time for the dynamic programming step for the subgraph

isomorphism based upon a tree decomposition. The latter takes $O(k^{t+1}n)$ time, where t, usually not >4, is the tree width of the tree decomposition and n is the number of stems in the structure. Recall that k is the number of candidates selected for the simplified CM model of a stem during the preprocessing; it is a relatively small parameter that can be used to tune the accuracy of the alignment. The time for the preprocessing step is $O(R^2Mn)$, where M is the maximum size of a CM and R is the maximum length of the sequence regions from which candidates are selected. These regions are fairly restricted by the preprocessing techniques we introduced here (Section 2.3). Our experiments showed that the preprocessing time $O(R^2Mn)$ is roughly the same as the time $O(k^{t+1}n)$ needed for the dynamic programming step when k is around 10 and that it is dominated by the latter for larger values of k or t. So the time for searching a whole genome is very much scalable with the size and complexity of the RNA structure searched.

Overall, our results indicate that the RNA graph model plus tree decomposition method incorporated into RNATOPS performed very well in efficiency while maintaining high search accuracy. The advantage of RNATOPS in speed, compared to other programs, increased as the length of the modeled molecule increased. This is because its search time depends on the number of stems, not the number of nucleotides, in the structure. Thus, the efficiency advantage becomes even more significant for RNATOPS to search for the larger yeast telomerase RNAand bacterial 16S rRNA (Tables 4.4 and 4.5 in Supplementary Material). Note that RNATOPS search accuracy can be tuned by the user through parameter k, to balance search sensitivity versus running time. The problems that RNATOPS had, where target RNAs were not found, were in stem-loop regions of tmRNAs where individual molecules deviated from the consensus structure; increasing the k value allowed RNATOPS to resolve most of these, at the cost of a slight decrease in speed.

4.6 DISCUSSION

Heuristic techniques have been presented in this article with the aim to develop a fast and accurate RNApseudoknot search program based on our previous work in an RNA graph modeling method. Through search tests on the implemented program RNATOPS, we have shown its performance comparable with or better than that of Infernal and FastR in identifying large or complex RNA structures including pseudoknots. We discuss in the following the strengths and weaknesses of RNATOPS.

One apparent advantage of RNATOPS is its ability to detect pseudoknots accurately without compromising computation time. Theoretically, RNATOPS can feasibly consider all combinations of stems for pseudoknot alignment through a non-conventional, tree decomposition-based dynamic programming. Detecting a pseudoknot as a whole structure avoids the difficulty with pseudoknot-free models that the predicted alignment sometimes incorrectly forms pseudoknot-free substructures in a 'pseudoknot territory'.

Another advantage of RNATOPS is its search speed. The theoretical time $O(k^{t+1}n)$ for structure–sequence alignment with RNATOPS has been effectively speeded up by the introduced heuristic techniques that can yield small values for k and t. Another important factor contributing to the efficient time is parameter n, the number of stems, not the number of nucleotides in the structure which would otherwise be at least one magnitude larger. As shown in the test results, RNATOPS has essentially broken the inefficiency barrier that might have heldback other pseudoknot detection models, reducing the computation time from hours to minutes.

Nevertheless, since the introduced heuristics produce only k pairs of candidate regions for each individual stem in the structure to align to, for a small k, they may not include the real

candidate of the stem and may bring inaccuracy to the search result. In particular, when a stem in the RNA contains non-canonical base pairings, for which candidates may not be accurately identified, it is possible that all pairs of candidates between this stem and another are 'incompatible', resulting in an invalid alignment and lower sensitivity. This issue does not exist in the CM–CYK-based programs like Infernal as its stem candidates are found globally instead of locally.

Another issue with the current version of RNATOPS is the computation of the structure–sequence alignment without reusing the data from the previous scanning window frame. In fact, the CM–CYK-based search method can save a factor of $O(M)$ computation time by reusing data between two consecutive window frames (Durbin et al., 1998), where M is the CM model length. This issue might have cost RNATOPS some speed in the search tests; however, we believe that it is possible to make technical improvements for RNATOPS in reusing the data between scanning window frames to further speed up the search.

We consider two future developments for RNATOPS. First, the graph model can also easily profile structures caused by nucleotide interactions beyond the binary base pairing. For example, the graph model makes it easy to profile tertiary interactions or triple helices recently found in the telomerase RNA genes of human and yeast genomes (Chen and Greider, 2004; Lin et al., 2004; Shefer et al., 2007; Theimer et al., 2005). Although one of the two stems involved in such a triple helix is actually formed by two base pairing regions that are arranged in the same direction ($5'$ to $3'$), our approach will allow the stem to be modeled with an individual CM the same way as modeling a regular stem, without the need of additional, new techniques.

Second, the current implementation of program does not allow the search for an instance of ncRNA in the target genome that differs in structure significantly from those in the training set;

nor can the current program consider alternative or optional substructures in RNAs. One solution to this will be to develop probabilistic profiling of variable substructures that may occur in the structure model. In particular, our modeling method makes it possible to characterize and implement the structure of an RNA family with a graph model that contains probabilistic edges to specify variable substructures. This will bear similarity to earlier methods by Holmes (2004) and Rivas (2005) but with the ability to include pseudoknots.

4.7 ACKNOWLEDGEMENTS

The authors are grateful to the anonymous referees for their constructive comments that have greatly improved this manuscript. We also thank the authors of Infernal, RSEARCH and FastR who have made their software packages publicly available to use.

Funding: National Institutes of Health (BISTI grant No: R01GM072080-01A1).

Conflict of Interest: none declared.

4.8 REFERENCES

[1] Bafna,V. and Zhang,S. (2004) FastR: fast database search tool for non-coding RNA. In Proceedings of the 3rd IEEE Computational Systems Bioinformatics Conference. Imperial College Press, London, pp. 52–61.

[2] Brown,J.W. (1999) The Ribonuclease P database. Nucleic Acids Res., 27, 314.

[3] Brown,M and Wilson,C. (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In Hunter,L. and Klein,T. (eds) Proceedings of Pacific Symposium on Biocomputing. World Scientific Publishing Co, Singapore.

[4] Cai,L. et al. (2003) Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics, 19 (Suppl. 1), i66–i73.

[5] Chen,L. and Greider,C.W. (2004)An emerging consensus for telomerase RNAstructure. Proc. Natl Acad. Sci. USA, 101, 14683–14684.

[6] Cole,J.R. et al. (2007) The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. Nucleic Acids Res., 35, D169–D172.

[7] Dandjinou,A.T. et al. (2004)Aphylogenetically based secondary structure for the yeast telomerase RNA. Curr. Biol., 14, 1148–1158.

[8] Durbin,R. et al. (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

[9] Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. Nucleic Acids Res., 22, 2079–2088.

[10] Frank,D.N. and Pace,N.R. (1998) Ribonuclease P: unity and diversity in a tRNA processing ribozyme. Annu. Rev. Biochem., 67, 153–180.

[11] Gautheret,D. and Lambert,A. (2001) Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. J. Mol. Biol., 313, 1003–1011.

[12] Griffiths-Jones,S. (2007) Annotating noncoding RNAgenes. Annu. Rev. Genomics Hum. Genet., 8, 279–298.

[13] Griffiths-Jones,S. et al. (2003) Rfam: an RNA family database. Nucleic Acids Res., 31, 439–441.

[14] Griffiths-Jones,S. et al. (2005) Rfam: annotating non-coding RNAs in complete genomes. Nucleic Acids Res., 33, D121–D124.

[15] Harris,J.K. et al. (2001) New insight into RNase P RNA structure from comparative analysis of the archaeal RNA. RNA, 7, 220–232.

[16] Hofacker,I.L. (2006) RNAs everywhere: geonom-wide annotation of structured RNAs. Genome Inform., 17, 281–282.

[17] Holmes,I. (2004) A probabilistic model for the evolution of RNA structure. BMC Bioinformatics, 5, 166.

[18] Infernal: inference of RNA alignments. (2008) http://infernal.janelia.org/ (last accessed date June 30, 2008).

[19] Klein,R.J. and Eddy,S.R. (2003) RSEARCH: finding homologs of single structured RNA sequences. BMC Bioinformatics, 4, 44.

[20] Lin,J. et al. (2004) A universal telomerase RNA core structure including structured motifs required for binding the telomerase reverse transcriptase protein. Proc. Natl Acad. Sci. USA, 101, 14713–14718.

[21] Liu,C. et al. (2006) Efficient annotation of non-coding RNA structures including pseudoknots via automated filters, In Proceedings of Life Science Society Computational Systems Biology Conference (CSB 2006). Imperial College Press, London, pp. 99–110.

[22] Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res., 25, 955–964.

[23] Moore,S.D. and Sauer,R.T. (2007) The tmRNA system for translational surveillance and ribosome rescue. Annu. Rev. Biochem., 76, 101–124.

[24] Nameki,N. et al. (1999) Functional and structural analysis of a pseudoknot upstream of the tag-encoded sequence in E. coli tmRNA. J. Mol. Biol., 286, 733–744.

[25] Nguyen,V.T. et al. (2001) 7SK small nuclear RNA binds to and inhibits the activity of CDK9/cyclin T complexes. Nature, 414, 322–325.

[26] Rivas,E. (2005) Evolutionary models for insertions and deletions in a probabilistic modeling framework. BMC Bioinformatics, 6, 63.

[27] Rivas,E. and Eddy,S.R. (1999) A dynamic programming algorithm for RNA structure prediction including pseudoknots. J. Mol. Biol., 285, 2053–2068.

[28] Rivas,E. and Eddy,S.R. (2000) The language of RNA: a formal grammar that includes pseudoknots. Bioinformatics, 16, 334–340.

[29] Rivas,E. and Eddy,S.R. (2001) Noncoding RNA gene detection using comparative sequence analysis. BMC Bioinformatics, 2, 8.

[30] Rivas,E. et al. (2001) Computational identification of noncoding RNAs in E. coli by comparative genomics. Curr. Biol., 11, 1369–1373.

[31] Robertson,N. and Seymour,P.D. (1986) Graph minors II. Algorithmic aspects of treewidth. J. Algorithms, 7, 309–322.

[32] Shefer,K. et al. (2007) A triple helix within a pseudoknot is a conserved and essential element of telomerase RNA Mol. Cell Biol., 27, 2130–2143.

[33] Song,Y. et al. (2005) Tree decomposition based fast searching for RNA structures with and without pseudoknots. Proc. IEEE Comput. Syst. Bioinform. Conf., IEEE Computer Society Press. 223–234.

[34] Song,Y. et al. (2006) Efficient parameterized algorithms for biopolymer structuresequence alignment. IEEE/ACM Trans. Comput. Biol. Bioinform., 3, 423–431.

[35] Theimer,C.A. et al. (2005) Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. Mol. Cell, 17, 671–682.

[36] Uemura,Y. et al. (1999) Tree adjoining grammars for RNA structure prediction. Theor. Comput. Sci., 210, 277–303.

[37] Washietl,S. et al. (2005) Fast and reliable prediction of noncoding RNAs. Proc. Natl Acad. Sci. USA, 102, 2454–2459.

[38] Weinberg,Z. and Ruzzo,W.L. (2004) Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. Bioinformatics, 20 (Suppl. 1), I334–I341.

[39] Weinberg,Z. and Ruzzo,W.L. (2006) Sequence-based heuristics for faster annotation of non-coding RNA families. Bioinformatics, 22, 35–39.

[40] Yang,Z. et al. (2001) The 7SK small nuclear RNA inhibits the CDK9/cyclin T1 kinase to control transcription. Nature, 414, 317–322.

[41] Zhang,S. et al. (2005) Searching genomes for noncoding RNAusing FastR. IEEE/ACM Trans. Comput. Biol. Bioinform., 2, 366–379.

4.9 SUPPLEMENTARY MATERIAL

This supplementary material contains the figures and tables referenced in the paper.

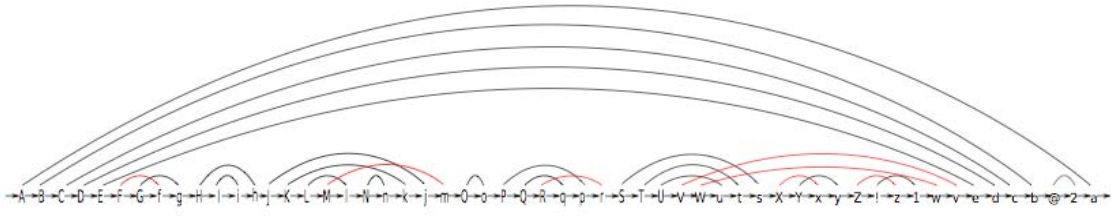


Figure 4.1 The structure graph for the consensus structure of bacterial tmRNAs, which contain four pseudoknots. Each stem is defined by an arc linking the upper and lower cases of the same letter. The red arcs indicate those stems causing pseudoknots.

```
>}
MMMMMMMLLLLLJJJ.BBBBBBjjjllllll..mmmmmmm.......bbbbbb
>BX248356
```

```
UGUCAGCCUAGA-UUCGUCUCUGG-UUUAGUGUCUGGCAUCGAUUAAGAGAC
>BX927150
UGUCAGCCUAGGGAA-GUCCCUGA-CCUAGAUCCUGGCAUCGACUAAGGGAC
>BA000035
UGUCAGCCCGGGGAU-GUCCCUGC-CCCGGAUGCUGGCAUCGACUAAGGGAC
>AY911523
UGUCAGUCCGGGUUC-GCCCUCGG-CCCGGGUACUGGCAUCAGCUAGAGGG-
```

Figure 4.2 The pasta representation for the second pseudoknot in the consensus structure of bacterial tmRNAs with some example sequences structurally aligned. The >} in the first line indicates that the line following it is a structure label (pairing) line; the sequences follow in fasta format.

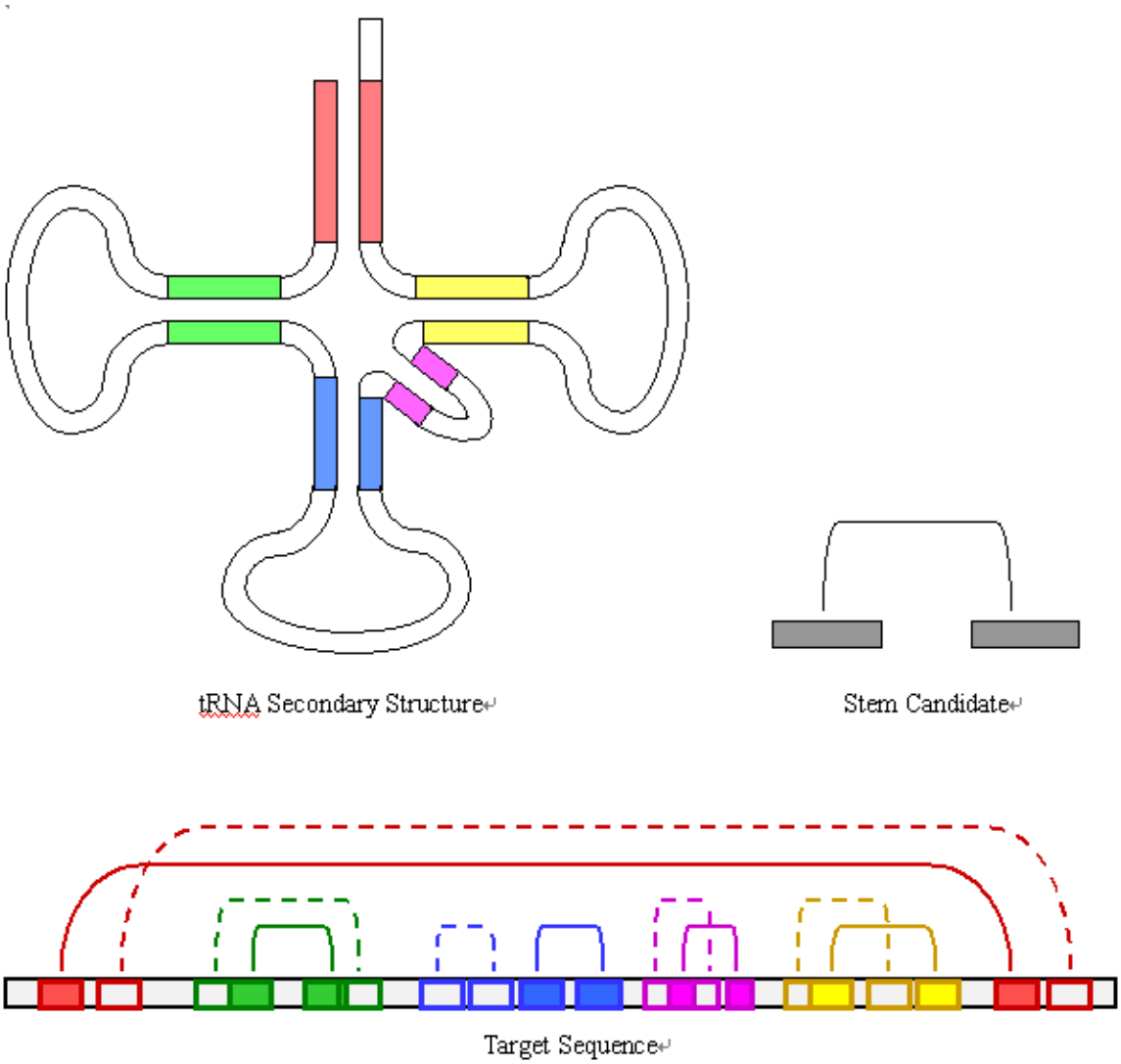
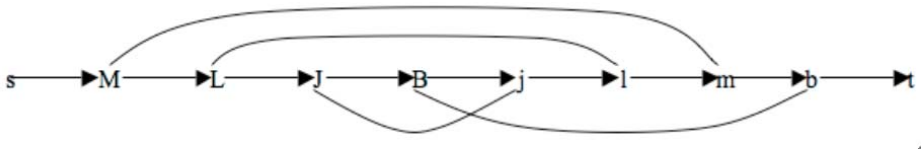


Figure 4.3 Schematic illustration for the preprocessing step to identify the top k candidates for each of the four stems of the tRNA consensus structure. Two candidates have been found for every stem in the target sequence, represented by two pairs of (possibly overlapping) rectangles (one pair hollow and the other solid) with the same color used in the tRNA structure.

(a) Structure graph for the second pseudoknot of the consensus structure of bacterial tmRNAs, where vertices s and t are added for technical purposes.



(b) The structure graph in (a) after removing arc (B, b) that crosses with the most other arcs.

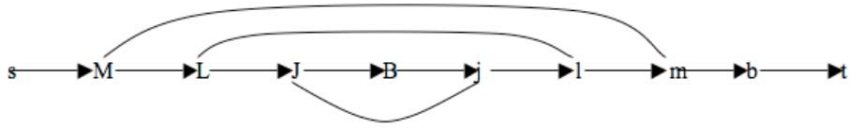
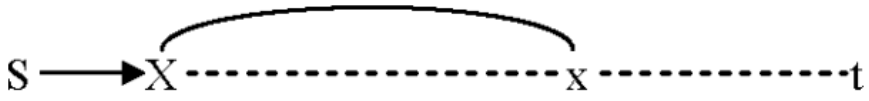
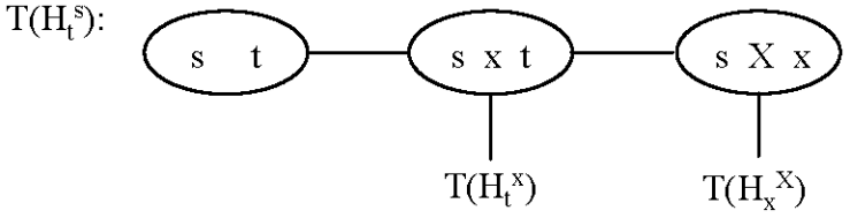


Figure 4.4

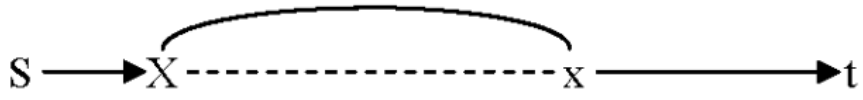
(a) Recursive case 1 for H_t^s : when (x, t) is not a directed edge.



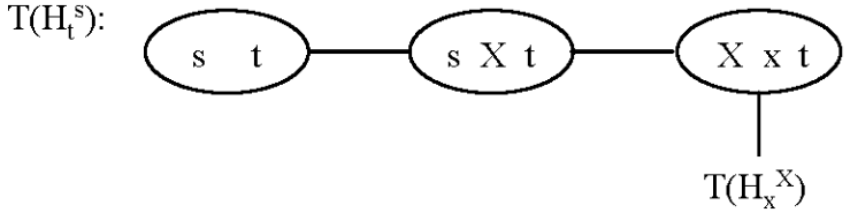
(b) Tree decomposition for (a).



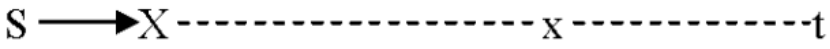
(c) Recursive case 2 for Hts: when (x, t) is a directed edge.



(d) Tree decomposition for (c).



(e) Recursive case 3 for Hts: when (X, x) is not an arc.



(f) Tree decomposition for (e).

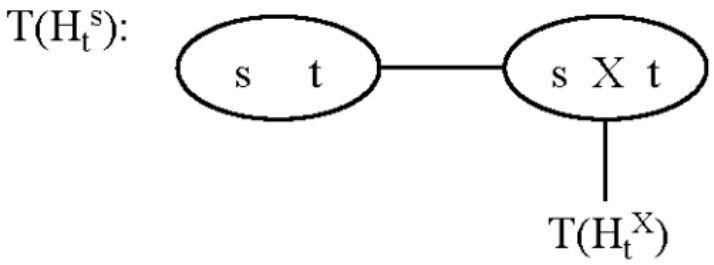


Figure 4.5 Recursive cases for the specialized tree decomposition algorithm. $T(H_t^s)$ denotes the tree decomposition for subgraph H_t^s . Node {s, t} is the root.

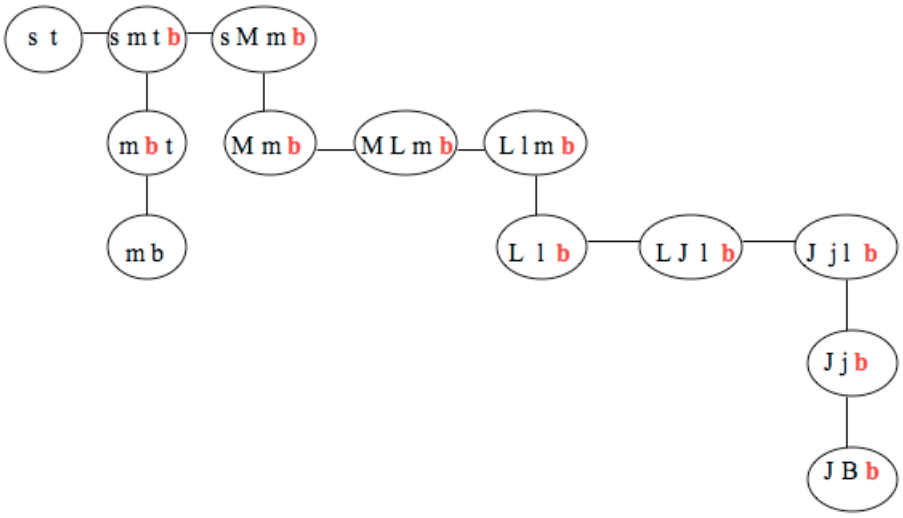


Figure 4.6 Tree decomposition for the structure graph shown in Figure 4.4 (a). It is obtained by first applying the specialized tree decomposition algorithm on the graph in Figure 4.4 (b) and then adding the vertex b (denoted with red, bold letter b) to every tree node on the path from the node containing b to the node containing B. The tree width is 3. Node {s, t} is the root.

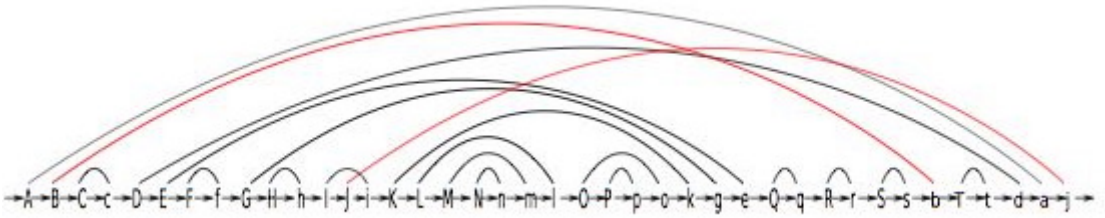


Figure 4.7 RNaseP (bacterial B) RNA structure graph

Table 4.1 tmRNA search results and comparison between RNATOPS and Infernal.

	RNATOPS			
	k=10	k=15	k=25	Infernal
Number of Training Sequences	42			42
Filter Used	HMM			HMM
Length of Filter	28			Not available
Number of Genomes Searched	43			43
Max/Min/Avg Genome Length (Mbps)	6.9/0.6/3.4			6.9/0.6/3.4
Max/Min of Time Used (Minutes)	9.3/0.04	13.1/0.12	39.2/0.3	18.8/2.2
Avg/Std of Time Used per Genome (Minutes)	7.9/1.2	11.8/1.6	35.6/4.1	10.7/4.5
Sensitivity	38/43	40/43	42/43	100%
Specificity	100%	100%	100%	100%

Table 4.2 tmRNA search results and comparison between RNATOPS and FastR.

	RNAToPS	FASTR
Number of Training Sequences	42 (364 bp)	online profile*
Filter Used	HMM	built-in
Filter Size	28 bp	not available
Number of Genomes Searched	43 (Length avg. 3.4M bp)	2* (4.5M bp & 0.6M bp)
Sensitivity	42/43 (k=25)	1/2*
Specificity	100%	100%
Time Per Genome	35 mins	58.3 hrs, 9.6 hrs (remote server)
Time Per M bp	11 mins	13.3 hrs (remote server)

*The FastR server includes tmRNAs as a pre-computed profile which may contain the RNA for the genome being searched. We selected one bacterial genome on which RNAToPS successfully

found the tmRNA, and another genome on which RNATOPS failed to find the tmRNA, then submitted these to the FastR server. FastR gave the same results as RNATOPS with these 2 sequences, finding the structure in one sequence in 58.3 hours and missing it in the other in 9.6 hours.

Table 4.3 RNaseP (Bact B) RNA search results and comparison between RNATOPS and Infernal.

	RNATOPS (K=10)	Infernal
Number of Training Seqs	9	9
Filter Used	HMM	HMM
Length of Filter	36	Not available
Number of Genomes Searched	7	7
Max/Min/Avg Genome Length (M bp)	5.1/1.8/3.1	5.1/1.8/3.1
Max/Min of Time Used (Minutes)	18.7/9.5	150.4/58.5
Avg/Std of Time Used per Genome (Minutes)	14.7/3.7	98/27.4
Sensitivity	100%	100%
Specificity	100%	100%

Table 4.4 Telomerase RNA search results and comparison between RNATOPS and Infernal.

	S. bayanus		*S. cerevisae*		*S. kudriavzevii*		*S. mikatae*	
	RNATOPS	Infernal	RNATOPS	Infernal	RNATOPS	Infernal	RNATOPS	Infernal
# of Training Sequences	5	5	5	5	5	5	5	5
Filter Used	HMM	HMM	HMM	HMM	HMM	HMM	HMM	HMM
Length of Filter	47	N/A	47	N/A	47	N/A	47	N/A
Genomes length (M bp)	9.96	9.96	11.9	11.9	10.4	10.4	10.5	10.5
Time Used (Minutes)	5.7	295.2	5.5	654.9	6.4	372.1	6.4	446.4
Sensitivity	100%	100%	100%	100%	100%	100%	100%	100%
Specificity	100%	100%	100%	100%	100%	100%	100%	100%

Table 4.5 16s rRNA search results and comparison between RNATOPS and Infernal.

	RNATOPS	Infernal
Number of Training Sequences	12	12
Filter Used	HMM	HMM
Length of Filter	111	Not Available
Number of Genomes Searched	11	11
Max/Min/Avg Genome Length (M bp)	5.1/2.6/4.0	
Avg/Std Time Used per Genome (Minutes)	14.1/2.4	88/18.42
Sensitivity	100%	100%
Specificity	100%	100%

Table 4.6 Structure alignment accuracy and comparison between RNATOPS and Infernal.

	tmRNA		RNaseP B RNA		Telomerase RNA	
	RNATOPS	Infernal	RNATOPS	Infernal	RNATOPS	Infernal
Number of Structures Correctly Found	40/43 (k=15)	43/43	7/7	7/7	4/4	44
Number of Found Structures with Stems off Position *	4/40	7/43	2/7	0/7	0/4	0/4
Total # of Stems off Position	9	17	4	0	0	0
Pseudoknot Regions Mistakenly Aligned **	0	5	0	0	0	0

* A stem was aligned to a position more than a few nucleotides away from its correct position.

**A supposedly pseudoknot region in the sequence was aligned to a pseudoknot-free substructure by mistake. This was caused by the CM-based method unable to deal with pseudoknots in a full scale.

CHAPTER 5

RNAV: NON-CODING RNA SECONDARY STRUCTURE VARIATION SEARCH VIA GRAPH HOMOMORPHISM[1]

[1]Huang,Z., Malmberg,R., Mohebbi, M and Cai,L. 2010. In Proceedings of Computational Systems Bioinformatics Conference (CSB 2010), August, 2010. Vol. 9, p. 56-69. Reprinted here with permission of publisher.

5.1 ABSTRACT

Non-coding RNA (ncRNA) secondary structural homologs can be detected effectively in genomes with profile-based search methods. However, due to the lack of appropriate ncRNA structural evolution models, it is difficult to accurately detect distant structural homologs, i.e., ncRNA structures with variations caused by evolutionary changes such as the insertion or deletion of a substantial portion in the structure. This paper presents results of an investigation toward developing a new framework for distant ncRNA structural homolog search. In this work, secondary structure conformations are modeled as graphs with small tree width and sequence-structure alignment for homolog detection is formulated as graph homomorphism. The technique of NULL stem is used to resolve the issue of optional stems that may be deleted from the structure profile or may be a misalignment. Test results on 51 benchmark data sets of Infernal (9 of them containing pseudoknots) show that a program based on these methods, RNAv, with the capability of detecting pseudoknots, has a comparable performance to the latest version of Infernal, and is better in detection of some distant homologs.

5.2 INTRODUCTION

Non-coding RNAs (ncRNA) are biologically important with functions in gene regulation, chromosome replication and RNA modification as well as other roles[10,24,36]. Homology-based searching methods[4,22,11,17,37,21,9] have become important for annotation of ncRNAs[12,14,22,28,29,33]. Genome search programs for ncRNA annotation have been developed[22,17,21,9] based on the covariance model (CM), a type of stochastic context-free grammar (SCFG), introduced by Eddy and Durbin[7]. The CM can profile position-specific compensatory mutations between base pairs as well as base conservations, yielding accurate ncRNA-specific and reconfigurable structural homolog search tools. Typically, the latest version of Infernal[9] can achieve more than 95%

accuracy in recognizing 51 benchmark ncRNA data sets with a high efficiency (Appendix, Table 5.1).

However, the structural search tools have been most successful in identifying ncRNAs homologs with little or small structural variation. RNA secondary and tertiary structures are both constant and variable across evolution[15,2,23]; that is, some sub-structures, such as stem-loops, will be found in all members of a given ncRNA family, but other sub-structures will be found only in some of the sequences of the family. Such structural variation poses novel challenges in profiling distant homologs for effective searches[23]. In previous work[18,19,25-27] RNA base and base pair evolution information was incorporated into SCFG models. To profile more substantial structural variations, usually these systems model variation with ribosomal RNA basepair evolution information due to the lack of more general, adequate structural evolution models. An improved model for RNA structural evolution has also been proposed[15,3] which can deal with limited degree of structure rearrangement between homologs but has yet to be incorporated into a search program. The program, trCYK[20], a local alignment algorithm for Infernal, contains a technical solution that addresses the issue of aligning the structural model with incomplete sequences. The scoring is based on conserved primary sequence and structure information instead of a structural evolution model. To date, a general method that addresses both possible misalignments and structural variation is still missing[1]. Searches for structurally distant homologs still largely rely on customized methods or tools[2].

The current paper reports preliminary results from our on-going effort in developing a profiling framework for effective search of ncRNA homologs that contain substantial structural variation. We profile the RNA secondary structure with the conformational graph model developed from a notion used in our previous work RNATOPS[16]. It is a coarse-grain model that

profiles the relationships (i.e., stems and loops) with graph vertices and edges. The current work is different from the previous research, however, in both search targets and supporting techniques. In particular, to detect structurally distant homologs, we describe the structural variation with novel graph homomorphism rules that can define the deletion/insertion of stems and loops with homomorphic mapping between an ancestor and a descendent structure graphs. The homomorphism rules allow deletion of edges and vertices from the conformational graph, which was not permitted in our previous work with RNATOPS. The detection of the structural variation is accomplished with a new technique of NULL stem that identifies any stem with a high probability of being deleted in the evolution. Although the threshold for such (evolutionary) probabilities is still being determined in a related study[32], the investigation of the graph homomorphic rules and implementation techniques is necessary because they are the mechanism to describe alternative and optional substructures, much the same role as context-free rules for CM[7].

We have tested on this new method to evaluate its capability to detect substructures (individual stems or combinations) possibly removed in the evolution. Typically, each used data set is a collection of multiple RNA sequences with a structural alignment and consensus, in which some stems may present in some but not all involved sequences. We have chosen to use the 51 benchmark data sets used by Infernal[9] in our tests. Although certain regions in these data sets are highly conserved, overall it exemplifies substantial structural variation. For example, we obtained (through calculations) 19.57 as the averaged standard deviation of the sequence lengths in these data sets. Totally there are 5686 training sequences in these 51 benchmark datasets, and 540 of them have at least one stem absent, accounting for 9.5% of the total number of sequences. Since Infernal performs well on these benchmarks, the evaluation on our method with

comparison to Infernal is appropriate. We conducted tests based on filtering method and non-filtering method, and compared the search results based on different ratio threshold for the percentage of the hit region overlapping with the real RNA region.

5.3 METHOD

We introduce a new method to profile RNA secondary structure variation for distant homolog search. It consists of three parts: the model to profile the consensus structure, rules for structural variation, and an algorithm to implement structure-sequence alignment and search. The model is based on the notion of conformational graph developed in our previous work to profile the consensus structure of multiple RNA sequences[16,30,31], with additional elements convenient for the description of structural variation.

5.3.1 CONFORMATIONAL GRAPH

The consensus secondary structure of RNA can be viewed as a topological relation among basic structural units, each of which is a stem or a loop. The structure model consists of two components: a weighted conformational graph that represents the relationship among all these basic structural units, and a set of simplified CMs and profile HMMs, each modeling a stem or a loop.

In such a conformational graph H, each vertex represents a contiguous sequence segment, either a loop or one of the two half-stems. It is a mixed graph containing both directed and undirected edges. Each directed edge connects two neighboring sequence segments, i.e. one of base-pairing stem regions and one loop region, and each undirected edge connects two base-paired sequence segments that form a stem. Fig.5.1 shows one example of a pseudoknot structure and the corresponding conformational graph, H (Fig.5.1(a)).

Searching in a target genome for a profiled structure consists of sliding a window of appropriate size along the target genome and aligning the structure model to a target sequence. Technically, the sequence segment within each window is preprocessed to identify the top k candidates for all CM models. Given the set of candidates of all profiled stems in the structure, a candidate graph can be constructed similarly to the construction of a conformational graph. Based on this construction, each vertex u in the conformation graph can only be mapped to a specific set of the same number of vertices in the candidate graph G, each of which is called a candidate of the vertex u.

5.3.2 HOMOMORPHISM FOR STRUCTURAL VARIATION

The optimal structure-sequence alignment between the structure model and the target sequence thus corresponds to finding, in the candidate graph, a maximum weighted subgraph that is homomorphic to the conformational graph. The weight is defined by the alignment score between vertices (stems and loops), in the conformational graph, and their counterparts, in the candidate graph. This graph homomorphism problem is an NP-hard problem[13], but tree decomposition based dynamic programming allows achieving efficiency for the computation[16].

To handle structure variations, the deletions allowed on the profile graph H can be classified into the following two categories.

(a) Deletion of a stem, which removes the base pairing between the two involved sequence segments. As shown in Fig.5.1(b), stem (3, 7) will be deleted from profile graph H. The homomorphic mapping merges vertex 3 (i.e., one arm of the stem), and its neighboring vertices, 2 and 4 (i.e., both loops), into one vertex, 3’(i.e. a loop). Similarly, it merges vertex 7, and its neighboring vertices, 6 and 8, into one vertex, 7’, in query sequence, Q.

(b) Deletion of a substructure,consisting of more than one stem, e.g., a pesudoknot structure, by applying (a) repeatedly. As shown in Fig.5.1(c), the pseudoknot structure contains stem (3, 7) and stem (5, 9) and loop 4, 6 and 8. To delete this pseudoknot structure, the first step is to delete stem (3, 7), which is to merge vertices 2, 3, 4 into vertex 3' and to merge vertices 6, 7, 8 into vertex 7'. The second step is to delete stem (5, 9), which is to merge 3', 5, 7' into vertex 5' and merge vertices 9, 10 into vertex 9' since 7' has been used.

The homomorphism used in this work is somewhat non-standard, as all operations need to meet the standard definition of graph homomorphism on vertices representing base-pairing regions only and not for vertices representing loops. In particular, edge preserving properties through homomorphism only apply to edges formed between vertices that represent base pairing regions and contribute to stems.

5.3.3 STRUCTURE-SEQUENCE ALIGNMENT

An alignment between a structure profile and a target sequence is essentially a homomorphism between the conformational graph H for the structure profile and some subgraph of the candidate graph G constructed from the target sequence. Generally, RNAv follows the basic idea of tree decomposition-based dynamic programming to compute the optimal alignment between graph H and the subgraph of G[16]. To consider structural variation, one special stem candidate, NULL stem, will be added to the candidates of every stem model in the profile. For each tree node, the algorithm examines all possible combinations of the candidates including the NULL stem candidate, from the number 0 to the number max_NULL_stem, in the tree node (where max_NULL_stem is the maximum number of NULL stems). Thus, the optimal alignment will consider all k+1 candidates for every stem in the tree decomposition based dynamic programming. For each tree node, the optimal alignment score and the number of NULL stems

(which can be technically constrainted) will be saved. The final optimal alignment score will be obtained in the tree root and a recursive process can be applied to trace back the optimal alignment. In this way, RNAv places a limit to the maximum number of NULL stems, max_NULL_stem, making it possible to identify from the target genome to RNAs conforming to the profiled structure but with possible structural variation from the consensus.

5.4 EVALUATION

The newly introduced methods have been implemented into the search program, RNAv, which has been tested in different gcc version 3.4.6, 4.2.1 and 4.4.1. We collected Infernal's benchmark dataset (51 ncRNA families) from RFAM seed alignment database (release 9.1) and tested them on the following four programs: Infernal (1.0.2), trCYK (Infernal can be accessed from http://infernal.janelia.org/ and trCYK is one of Infernal's functions), RNAv and RNATOPS, where trCYK is a new function of Infernal for local alignment to search for structure on incomplete query sequence, and RNATOPS is an earlier version of our program that allows little and small structural variation. This section will evaluate the performance of RNAv using Infernal's performance as a reference. Due to page limitation, we have to move some of tables and figures to the appendix. We also created a webpage (http://www.cs.uga.edu/~zhibin/csb2010_RNAv_data.html) containing all the tested data results for this paper.

5.4.1 DATA PREPARATION AND EVALUATION CRITERIA

Infernal's benchmark datasets do not contain any sequence pair that is more than 60% identical[8,9]. Each data set is a multiple structure alignment including the annotation of the consensus structure. We used each data set as training data to construct a structure profile for search. For the purpose of testing the recognition capability, we designed the following leave-one-out,

pseudo-genomic searches: we followed a cross-validation approach and embedded each RNA sequence, which was removed from the training alignment, in the middle of a 2000-nucleotide-long random sequence, which shares the same nucleotide frequency as that RNA sequence. The remaining alignment sequences were used as the training set for a search on that pseudo-genome. We applied both an HMM filtering method[34, 35] (Infernal also uses QDB-filtering method[8] and the non-filtering method to the pseudo-genome test.

There are two levels of search performance. The first level is to compare the predicted position of the tested RNA with its real position on the searched genome. The second level is to compare the predicted structure with its real structure. For position comparison, we used the percentage ratio of overlap, between the real RNA sequence and the predicted one, with different thresholds (0.75, 0.8, 0.85, 0.9 and 0.95). Position performances of these search programs with these different thresholds are shown in Appendix-Fig. 5.2. In this section, we analyzed the results with the threshold of 0.85.

In these 51 datasets, 9 of them contain pseudoknots. Since Infernal does not explicitly predict pseudoknot structures, we remove the crossing stems from those pseudoknot structures when testing them on Infernal and trCYK. On all programs, the top one hit candidate reported was taken as the prediction.

5.4.2 POSITION SEARCH ACCURACY

The search position accuracy comparisons between Infernal, trCYK and RNAv are shown in Fig. 5.3 and Appendix-Table 5.1. Infernal has the highest average position prediction performance, 97.51% using the filtering method and 97.67% in the non-filtering method. RNAv gets 93.70% in the filtering method and 93.73% in the non-filtering method, followed by trCYK, which gets the accuracy, 89.28%. However, in 10 datasets, RNAv's filtering-search appears to perform

better than Infernal and in 7 datasets its non-filtering-search performance is better than Infernal. We focus on analysis between the results of Infernal and of RNAv as trCYK, local motif search function, may not be entirely appropriate for detecting global structure with missing substructures.

5.4.3 CAPABILITY TO DETECT STRUCTURAL VARIATION

We analyze the capability of RNAv in detecting structural variation by examining those cases that missed by Infernal. There are 10 such datasets, for which RNAv's filtering search performance was better than Infernal and 7 datasets, for which RNAv's non-filtering search performance is better than Infernal (we labeled these 17 dataset in bold font in Appendix-Table 5.1). Due to the page limitation, we picked 4 structure prediction typical cases to analyze: RF00023(Bacterial tmRNA) from the filtering-search test, and RF00024(Telomerase-vert), RF00029(Intron_gpII) and RF00230(T-box), from the non-filtering-search test.

5.4.3.1 RF00023 BACTERIAL TMRNA

RF00023, Bacterial tmRNA, has 228 training sequences, and the length of sequences in this alignment file ranges from 235 to 436, and its standard deviation is 26.35. We also calculated the pseudo-energy score for all the stems, and used the threshold of -4.0 to estimate, in the original alignment file, the number of good/NULL/weak stems (Appendix-Table 5.3).

Test result (Appendix-Table 5.2) shows that, in this dataset, RNAv found all of stems, and Infernal missed 4 cases (with the index of sequence 98/212/219/225 in the alignment file). Checking the 4 cases Infernal missed revealed that most of interior stems in these 4 alignments are weak stems while the outer stems are good stems.

We also calculated the number of NULL stem in the original alignment and candidate hit alignment (Appendix-Table 5.4, Table 5.5). There are 77 NULL stems in this RF00023

alignment file, and RNAv finds 30 of them. Actually RNAv detects 216 NULL stems, 30 of them are real NULL stem in the original alignment, and 186 of them were used to predict weak stems that are 100 in number (Here we used -4.0 as the threshold of pseudo-energy score to determine weak stems). For the other 47 real NULL stems, RNAv detected real stems for them and 25 of them are good stems. We show the test result of the 98th query sequence as an example to explain RNAv's performance discussed here.

In this case, RNAv detected stem N/L/E/M and NULL stem H correctly (Fig. 5.5); for stem A/B/C/I/D, actually in the original alignment these stems are very weak (Fig. 5.4), containing many non-canonical base-pairings, and RNAv predicted the candidates with lower pseudo-energy score; for stem K/J/F/G, RNAv could not find the candidates for them and used NULL stems to represent these two stems (see Appendix-Tables 5.4 and 5.5).

5.4.3.2 RF00024 TELOMERASE-VERT

RF00024, Telomerase-vert, has 37 training sequences. The length of sequences in this alignment file ranges from 382 to 559, and its standard deviation is 38.21. We also estimate, in the original alignment file, the number of good/NULL/weak stems (see Appendix-Table 5.6).

Test result (Appendix-Table 5.2) shows that, in this dataset, RNAv missed 1 case, and Infernal missed 2 cases. We checked those missed cases of RNAv and Infernal. For the one missed case, RNAv only detected part of the whole structure correctly, resulting in the overlap region not large than 85%. For those two missed cases, Infernal detected local hit, 12.23% and 27.7% of the whole structure hit respectively.

We compute the number of NULL stems in the original RF00024 alignment and candidate hit alignment (Appendix-Table 5.7, Table 5.8). Totally there are 26 NULL stems in this RF00024 alignment file, and RNAv finds 9 of them. The total number of NULL stems in the

candidate hits are 66, and RNAv uses 57 NULL stems to replace weak stems and 9 of them are real weak stems.

For example, in this test of the 22nd query sequence, there are total 17 stems and15 of them are real stems. RNAv found (Fig.5.7) those two NULL stems, Stem F/E correctly. For those 15 real stems, RNAv detected 12 of them correctly, and 3 of them mostly correct but with some nucleotides shifted. Infernal (Fig.5.8) found a candidate for Stem F/E, which was actually no sequence in the original alignment, and some nucleotides shifted in the candidate stem of K.

5.4.3.3 RF00029 INTRON_GPII

RF00029, Intron_gpII, has 113 training sequences. The length of sequences in this alignment file ranges from 61 to 154, and its standard deviation is 22.03. We also estimate, in the original alignment file, the number of good/NULL/weak stems (Appendix-Table 5.9).

Test result (Appendix-Table 5.2) shows that, in this dataset, RNAv missed 1 case, and Infernal missed 7 cases.

We now use the test of the 98th query sequence as an example to explain the performance difference between RNAv and Infernal. We checked the original alignment file and found there was a special stem C that had a big sequence variation within its loop region. RNAv and Infernal both predicted the first two stems, Stem A/B, correctly. For the last stem, Stem C, RNAv found one candidate stem with a lower pseudo-energy score than the real one, while Infernal found one candidate stem with a higher pseudo-energy score (Fig.5.9, Fig.5.10, Fig.5.11). However, RNAv failed in the one with the largest sequence variation. Infernal only outputted local structure search results for those 7 missed cases.

5.4.3.4 RF00230 T-BOX

RF00230, T-box, has 65 training sequences. The length of sequences in this alignment file ranges from 167 to 370, and its standard deviation is 32.86. We also estimate, in the original alignment file, the number of good/NULL/weak stems (Appendix-Table 5.12).

We checked the original alignment file and found there was a loop region, between Stem G and Stem H, which has a big sequence variation. RNAv missed 2 cases, outputting only local structure hits, so did Infernal in those missed 6 cases (Appendix-Table 5.2).

Here we analyzed the test result with the 26th query sequence. In this test, RNAv found most of the real stems correctly, and found Stem E with both sides having a position shift, and used a NULL stem to replace Stem C, which actually had high pseudo-energy score in the original alignment file. Interestingly, RNAv and Infernal both found the same candidate for Stem H, which was different the one in the original alignment, but Infernal could not find Stem I.

5.4.4 RNAV VS. RNATOPS

One of problems in RNATOPS is if heuristic preprocessing step does not include the real candidate of the stem in those k pairs of candidate regions for each individual stem, then it may fail[16]. Actually this was the original motivation of proposing NULL stem technique. We used RNATOPS to repeat RNAv's filtering search test, and see how much improvement RNAv can make using NULL stem technique. Test result shows that RNAv can improve about 16% of accuracy in filtering method and 13% in non-filtering method.

5.4.5 ANALYSIS OF RNAV'S PARAMETERS

There are two parameters in RNAv. One is k, the number of stem candidates; the other is max_NULL_stem, the maximum number of NULL stems. In general, the values of these parameters are determined by the training data. When sequences in the alignment are conserved,

a small value for k can yield decent search accuracy and larger values for k may further improve/fine-tune search results. On the other hand, if the data manifest some significant structural variation, the search accuracy may not be substantially improved by simply increasing values for k; while parameter, max_NULL_stem, affects the search result.

5.5 DISCUSSION

In this paper, we presented preliminary results from our on-going research in developing a new profiling framework for RNA secondary structure search for distant homologs. The new method profiles substantial structural variation with the conformational graph we previously developed; the newly introduced graph homomorphic mapping rules and the NULL stem technique make it possible to effectively detect substantial structure variation, typically stems missing in the structure because of evolution. Evident by the test results, the implemented program, RNAv, had comparable overall performance as Infernal on the 51 benchmark data sets selected and used for testing Infernal. RNAv was able to detect some structural variations that were missed by Infernal. Overall impression from the tests is that RNAv works for ncRNA search with diversed sequences while Infernal works with conserved ncRNA sequences. The comparison between RNAv with the earlier version RNATOPS shows an overall enhancement in performance, with more than 13% of accuracy improvement (Appendix, Fig. 5.2). The same table also shows the performance of trCYK, a new local alignment algorithm for Infernal that can locally aligns the structural model with incomplete sequences. Our result shows that local motif search techniques may not be entirely appropriate for detecting global structure with missing substructures.

In addition to the capability of handling pseudoknots and the search efficiency inherited from RNATOPS[16], there are a couple of more advantages demonstrated by RNAv. One is RNAv's capability to suppress some impact of noisy training data. Profile-based search

algorithms can be inherently alignment-sensitive. If more than 50% of a stem alignment contains canonical base pairs and others are non-canonical base pairs, then the stem modeling based on this alignment will be correct. When this correct model is used to predict those non-canonical base pairs, the score of searched stem candidates will be insignificant. In this scenario, RNAv may use NULL stem as the predicted local structure when all possible stem candidates are "too weak" to be meanful. This explains the reason why Infernal missed those 4 cases with interior weak stems and outer good stems in Bacterial tmRNA data set while RNAv found them.

Another interesting advantage of RNAv is its potential for detection of evolutionary structural changes. In testing the 51 data sets, RNAv was able to detect at least 34% or more regions with missing stems in each data set. These regions are presumably to have evolved to unpaired loop regions instead to base pairing stem regions or drastic mutations have caused stems in these regions to disappear. Therefore, RNAv may present as a technical solution to the issue of modeling stem evolution including insertion or deletion. One can apply RNAv to search for an ncRNA of interest across species, which may not be conserved in the structure, leading to the discovery of new members of the RNA, possibly in evolutionarily distant species.

Graph homomorphic mapping appears to be powerful to account for ncRNAs structure evolution. Together with the structure evolution study[35] on specific ncRNAs and the notion of graph homomorphic mapping to define stem insertion and deletion, RNAv and the underlying method will be further developed into an accurate solution to detecting distant structural homologs.

5.6 ACKNOWLEDGMENTS

This research project was supported in part by NIH BISTI R01GM072080-01A1 grant and NIH ARRA Administrative supplement to this grant. We also thank the authors of Infernal who have made their software packages publicly available to use.

Conflict of Interest: none declared.

5.7 REFERENCES

[1] Andreas R. Gruber, et al. 2010. Rnaz 2.0: Improved Noncoding Rna Detection, Pacific Symposium on Biocomputing 15:69-79.

[2] Axel Mosig. et al. (2009) Customized strategies for discovering distant ncRNA homologs, Briefings in Functional Genomics and Proteomics, doi:10.1093/bfgp/elp035.

[3] Bradley RK, Holmes I (2009) Evolutionary Triplet Models of Structured RNA. PLoS Comput Biol 5(8): e1000483. doi:10.1371/journal.pcbi.1000483.

[4] Brown,M and Wilson,C. (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In Hunter,L. and Klein,T. (eds) Proceedings of Pacific Symposium on Biocomputing. World Scientific Publishing Co, Singapore.

[5] Cai,L. et al. (2003) Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics, 19 (Suppl. 1), i66–i73.

[6] Durbin,R. et al. (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.

[7] Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. Nucleic Acids Res., 22, 2079–2088.

[8] Eric P. Nawrocki and Sean R. Eddy. (2007) Query-dependent banding (QDB) for faster RNA similarity searches. PLoS Comput. Biol., 3, e56.

[9] Eric P. Nawrocki, Diana L. Kolbe, and Sean R. Eddy, Infernal 1.0: inference of RNA alignments, Bioinformatics. 2009 May 15;25(10):1335-7.

[10] Frank,D.N. and Pace,N.R. 1998. Ribonuclease P: unity and diversity in a tRNA processing ribozyme. Annu. Rev. Biochem., 67, 153–180.

[11] Gautheret D., and A. Lambert. 2001. Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. J. Mol. Biol. 313:1003–1011.

[12] Griffiths-Jones,S. (2007) Annotating noncoding RNAgenes. Annu. Rev. Genomics Hum. Genet., 8, 279–298.

[13] Hell, Pavol; et al. (2004). Graphs and Homomorphisms (Oxford Lecture Series in Mathematics and Its Applications). Oxford University Press.

[14] Hofacker,I.L. (2006) RNAs everywhere: geonom-wide annotation of structured RNAs. Genome Inform., 17, 281–282.

[15] Holmes I: A probabilistic model for the evolution of RNA structure. BMC Bioinformatics 2004, 5:166.

[16] Huang, Z., Y. Wu, J. Robertson, L. Feng, R. Malmberg, and L. Cai. 2008. Fast and accurate search for non-coding rna pseudoknot structures in genomes. Bioinformatics. 24:2281–2287.

[17] Klein,R.J. and Eddy,S.R. (2003) RSEARCH: finding homologs of single structured RNA sequences. BMC Bioinformatics, 4, 44.

[18] Knudsen B, Hein J (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics 15: 446–454.

[19] Knudsen B, Hein J (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. Nucleic Acids Res 31: 3423–3428.

[20] Kolbe DL, Eddy SR, Local RNA structure alignment with incomplete sequence, Bioinformatics, 25(10):1236-1243, 2009.

[21] Liu,C. et al. (2006) Efficient annotation of non-coding RNA structures including pseudoknots via automated filters, In Proceedings of Life Science Society Computational Systems Biology Conference (CSB 2006). Imperial College Press, London, pp. 99–110.

[22] Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res., 25, 955–964.

[23] Menzel P, Gorodkin J, Stadler PF. (2009) The tedious task of finding homologous non-coding RNA genes. RNA, 15(12):2075-2082, 2009.

[24] Nguyen,V.T. et al. (2001) 7SK small nuclear RNA binds to and inhibits the activity of CDK9/cyclin T complexes. Nature, 414, 322–325.

[25] Pedersen JS, Meyer IM, Forsberg R, Simmonds P, Hein J (2004a) A comparative method for finding and folding RNA secondary structures within protein-coding regions. Nucleic Acids Res 32: 4925–4936.

[26] Pedersen JS, Forsberg R, Meyer IM, Hein J (2004b) An evolutionary model for protein-coding regions with conserved RNA structure. Mol Biol Evol 21: 1913–1922.

[27] Pedersen, J.S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E., Rogers, J., Kent, J., Miller, W., and Haussler, D. Identification and Classification of Conserved RNA Secondary Structures in the Human Genome. PLoS Computat Biol 2(4), e33 (2006).

[28] Rivas,E. and Eddy,S.R. (2001) Noncoding RNA gene detection using comparative sequence analysis. BMC Bioinformatics, 2, 8.

[29] Rivas,E. et al. (2001) Computational identification of noncoding RNAs in E. coli by comparative genomics. Curr. Biol., 11, 1369–1373.

[30] Song,Y. et al. (2005) Tree decomposition based fast search of RNA structures including pseudoknots in genomes. Proc. IEEE Comput. Syst. Bioinform. Conf., IEEE Computer Society Press. 223–234.

[31] Song,Y. et al. (2006) Efficient parameterized algorithms for biopolymer structuresequence alignment. IEEE/ACM Trans. Comput. Biol. Bioinform., 3, 423–431.

[32] Srivastava, A., Cai, L., Mrazek, J., Malmberg, R.L. 2010, Evolutionary Analysis of Bacterial tmRNA and RNAseP Secondary Structures, Research Article, in revision.

[33] Washietl,S. et al. (2005) Fast and reliable prediction of noncoding RNAs. Proc. Natl Acad. Sci. USA, 102, 2454–2459.

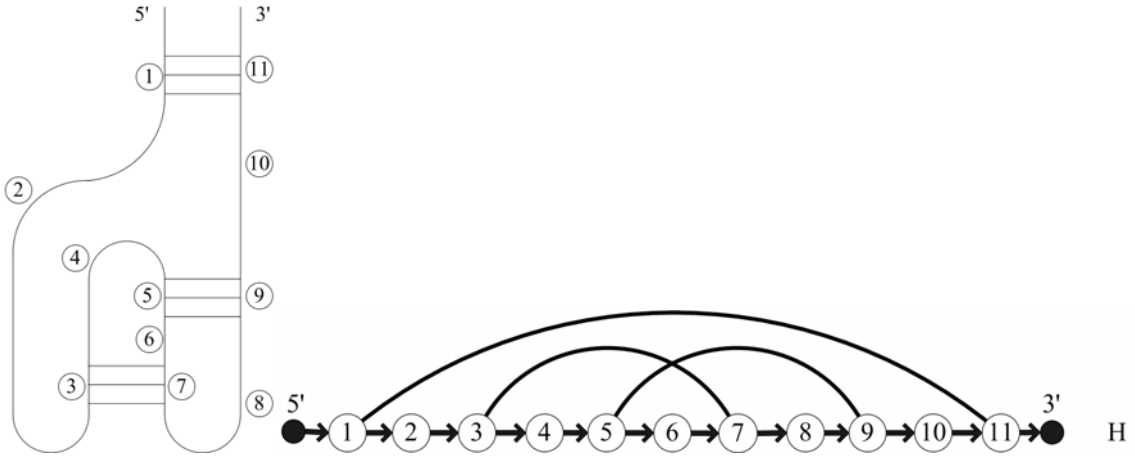
[34] Weinberg, Z. and Ruzzo, W. L. (2004). Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. Bioinformatics, 20 Suppl. 1:I334–I341.

[35] Weinberg, Z. and Ruzzo, W. L. (2006). Sequence-based heuristics for faster annotation of non-coding RNA families. Bioinformatics, 22:35–39.

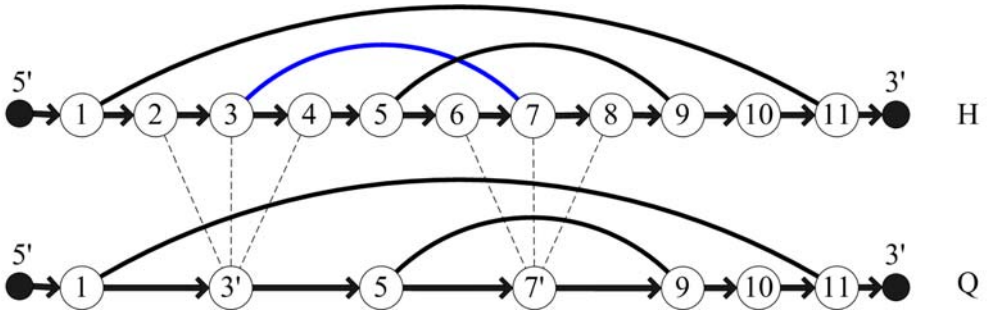
[36] Yang,Z. et al. (2001) The 7SK small nuclear RNA inhibits the CDK9/cyclin T1 kinase to control transcription. Nature, 414, 317–322.

[37] Zhang, S., B. Haas, E. Eskin, and V. Bafna. 2005. Searching genomes for noncoding RNA using FastR. IEEE/ACM Trans. Comput. Biol. Bioinform. 2:366–379.

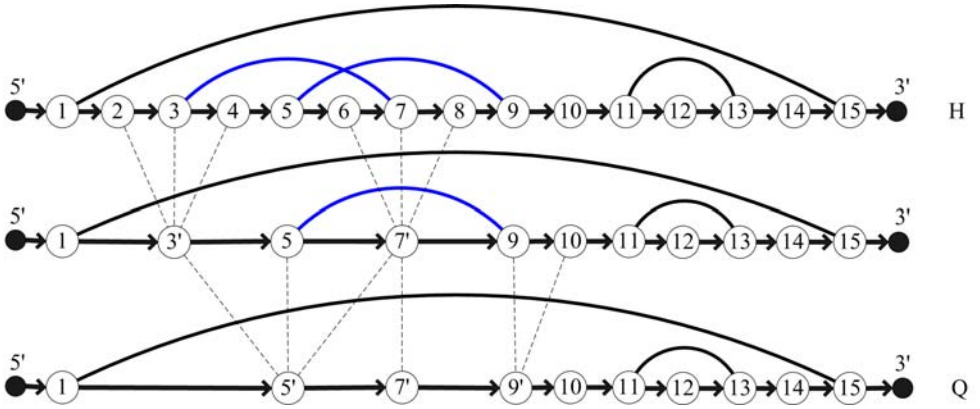
5.8 APPENDIX



(a) A pseudoknot structure and the corresponding conformational graph, H



(b) Graph homomorphism from H to query sequence, Q, which deletes stem(3, 7) from H.



(c) Graph homomorphism from H to query sequence, Q, which deletes one pseudoknot structure stem(3, 7)&stem(5, 9) from H.

Fig. 5.1. Graph homomorphism from H to query sequence Q.

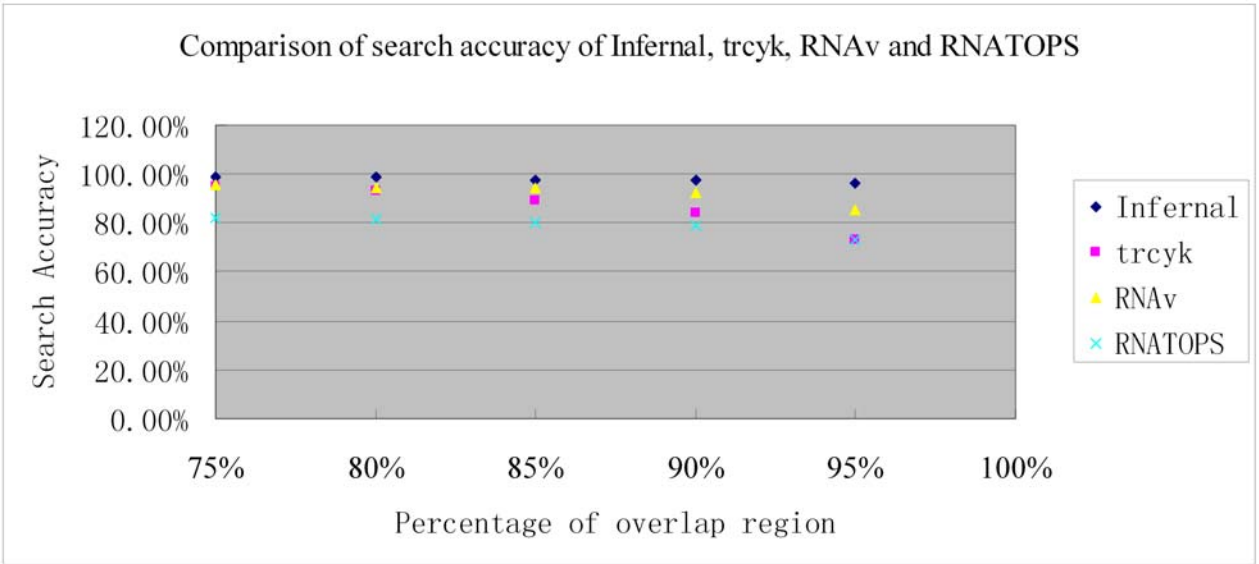


Fig. 5.2. Comparison of search accuracy of Infernal, trCYK, RNAv and RNATOPS in nonfiltering method

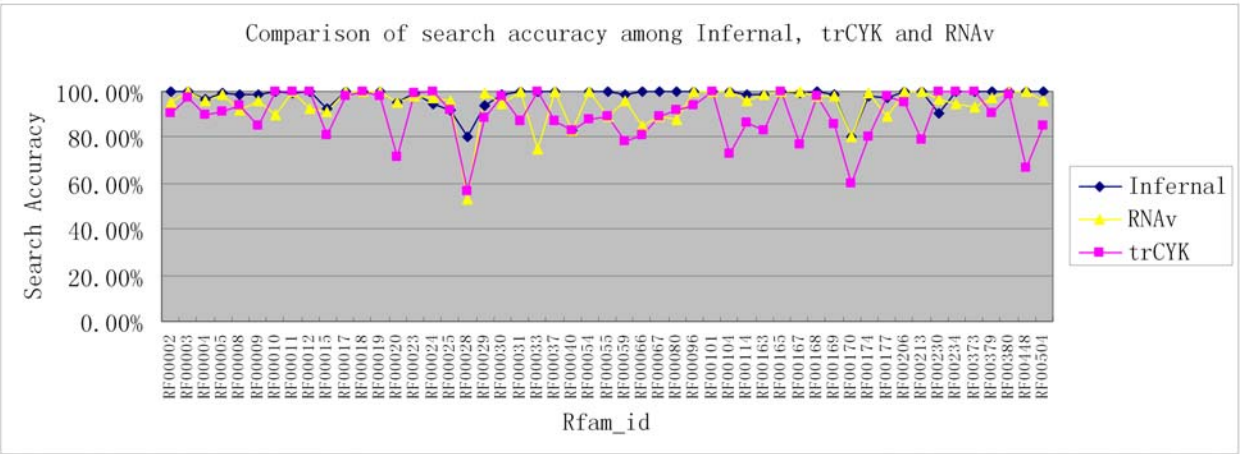


Fig. 5.3. Comparison of position prediction among Infernal, trCYK and RNAv.

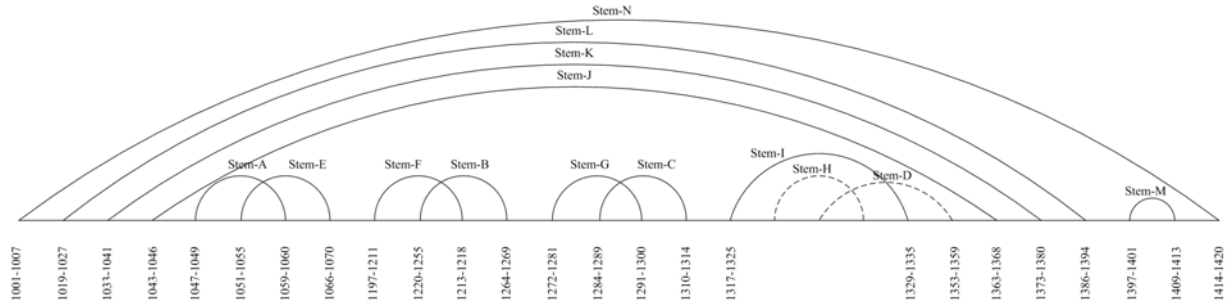


Fig. 5.4. Alignment of case-98 in the RF00023 dataset (x-axis is the position of nucleotide in the pseudogenome, arc with dash-line means NULL stem. Same for all other figures).

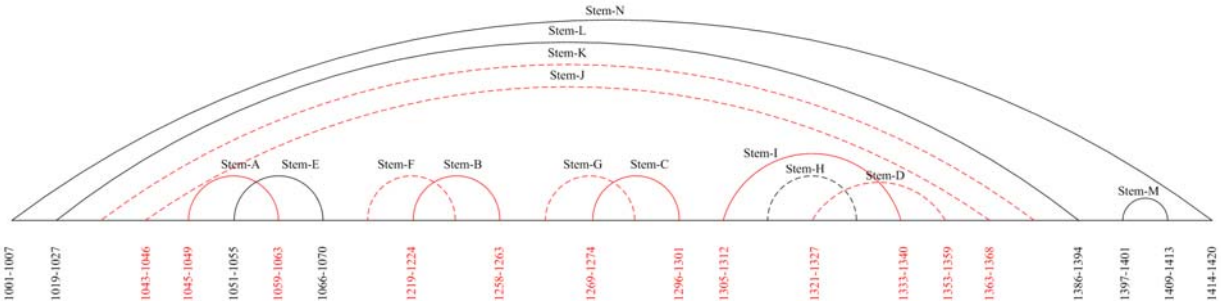


Fig. 5.5. Alignment of RNAv's result of case-98 in the RF00023 dataset, arc in red color means some difference from the original one. Same for all other figures.

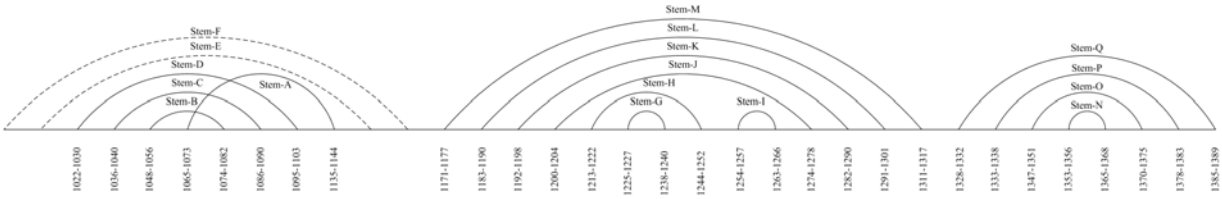


Fig. 5.6. Alignment of case-22 in the RF00024 dataset (x-axis is the position of nucleotide in the pseudogenome).

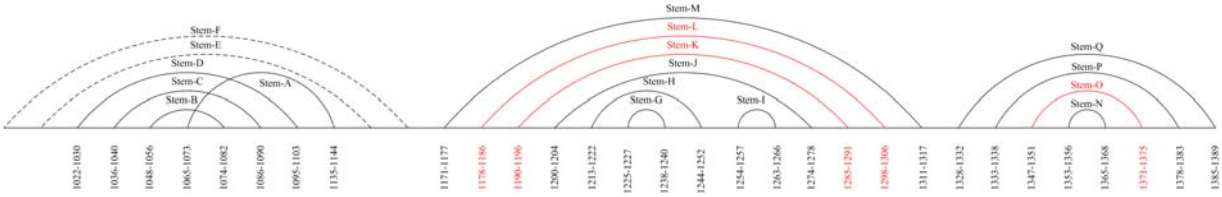


Fig. 5.7. Alignment of RNAv's result of case-22 in the RF00024 dataset.

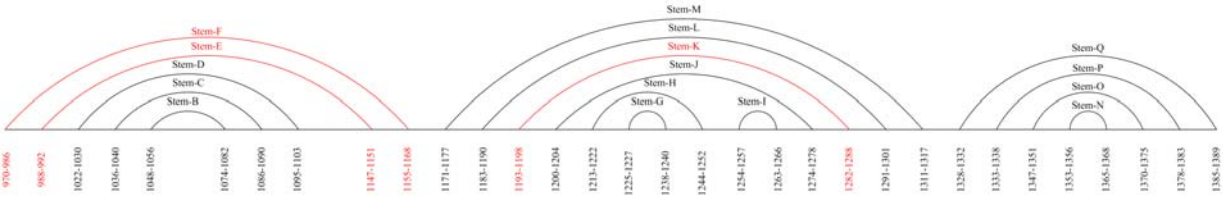


Fig. 5.8. Alignment of Infernal's result of case-22 in the RF00024 dataset.

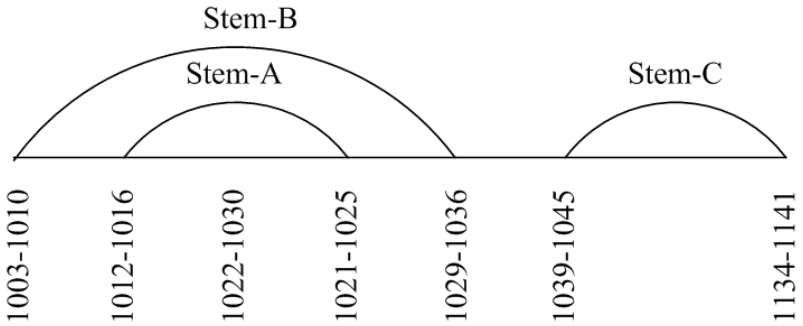


Fig. 5.9. Alignment of case-98 in the RF00029 dataset (x-axis is the position of nucleotide in the pseudogenome)

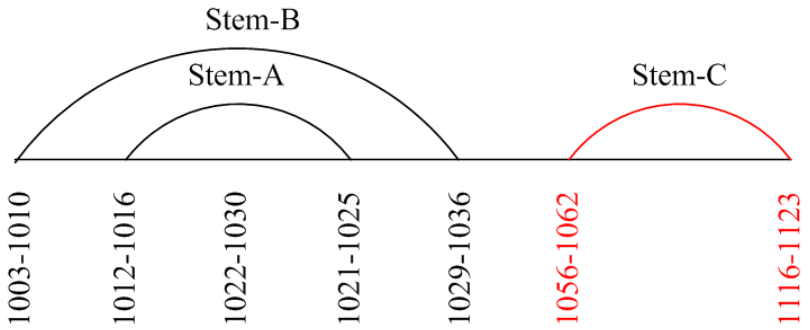


Fig. 5.10. Alignment of RNAv's result of case-98 in the RF00029 dataset

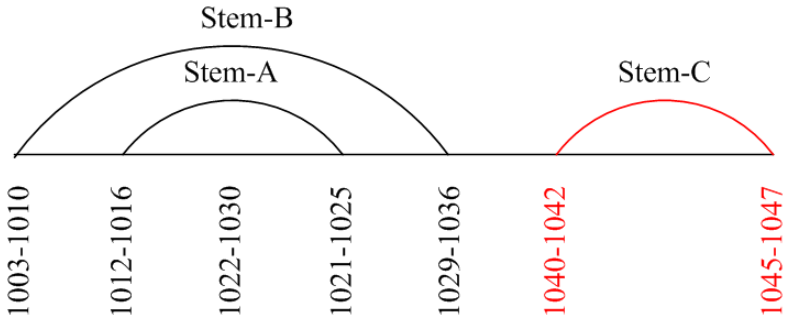


Fig. 5.11. Alignment of Infernal's result of case-98 in the RF00029 dataset

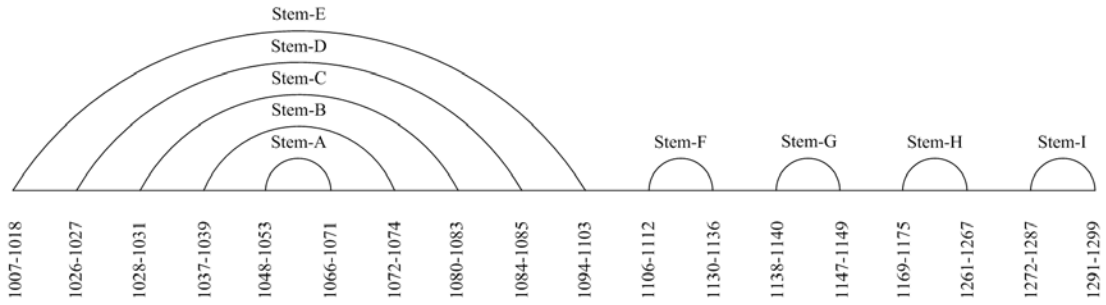


Fig. 5.12. Alignment of case-26 in the RF00230 dataset (x-axis is the position of nucleotide in the pseudogenome)

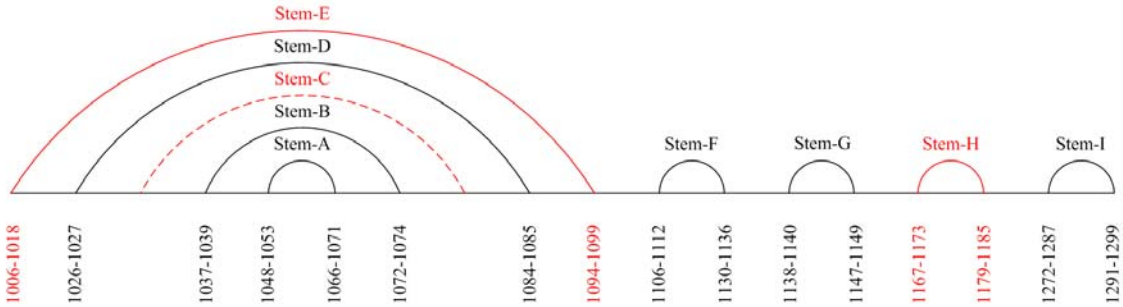


Fig. 5.13. Alignment of RNAv's result of case-26 in the RF00230 dataset

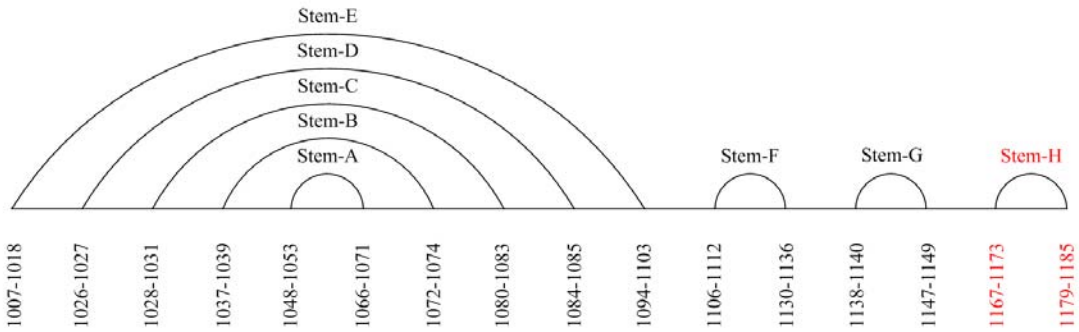


Fig. 5.14. Alignment of Infernal's result of case-26 in the RF00230 dataset

Table 5.1. Comparison of filtering/nonfiltering search accuracy among Infernal, trCYK and RNAv(ratio_threshold=0.85)

RFAM_id	RNA_name	No. of total training sequences	Infernal		with trCYK	RNAv	
			Filtering	NonFiltering		Filtering	NonFiltering
RF00002	5_8S_rRNA	62	100.00%	100.00%	90.32%	95.16%	95.16%
RF00003	U1	100	99.00%	100.00%	97.00%	97.00%	100.00%
RF00004	U2	212	96.70%	96.70%	90.09%	95.75%	96.23%
RF00005	tRNA	1052	95.91%	99.62%	91.16%	94.77%	98.48%
RF00008	Hammerhead_3	84	98.81%	98.81%	94.05%	86.90%	91.67%
RF00009	RNaseP_nuc	122	98.36%	98.36%	85.25%	95.08%	95.90%
RF00010	RNaseP_bact_a	306	100.00%	100.00%	99.67%	99.67%	89.87%
RF00011	**RNaseP_bact_b**	**115**	**99.13%**	**99.13%**	100.00%	**100.00%**	**100.00%**
RF00012	U3	27	100.00%	100.00%	100.00%	88.89%	92.59%
RF00015	U4	184	92.39%	92.39%	80.98%	89.67%	91.30%
RF00017	**SRP_euk_arch**	**104**	**97.12%**	100.00%	98.08%	**100.00%**	100.00%
RF00018	CsrB	14	100.00%	100.00%	100.00%	100.00%	100.00%
RF00019	Y	127	100.00%	100.00%	97.64%	100.00%	100.00%
RF00020	**U5**	**184**	**94.57%**	95.11%	71.74%	**95.11%**	95.11%
RF00023	**tmRNA**	**228**	**98.25%**	99.12%	99.56%	**100.00%**	98.25%
RF00024	**Telomerase-**	**37**	**91.89%**	**94.59%**	100.00%	**97.30%**	**97.30%**
RF00025	**Telomerase-cil**	**24**	**91.67%**	**91.67%**	91.67%	**100.00%**	**95.83%**
RF00028	Intron_gpI	30	80.00%	80.00%	56.67%	60.00%	53.33%
RF00029	**Intron_gpII**	**113**	**93.81%**	**93.81%**	88.50%	**98.23%**	**99.12%**
RF00030	RNase_MRP	89	93.26%	98.88%	97.75%	84.27%	94.38%
RF00031	SECIS	61	100.00%	100.00%	86.89%	100.00%	100.00%
RF00033	MicF	4	100.00%	100.00%	100.00%	75.00%	75.00%
RF00037	IRE	39	100.00%	100.00%	87.18%	92.31%	100.00%
RF00040	rne5	6	83.33%	83.33%	83.33%	83.33%	83.33%

RF00054	U25	8	100.00%	100.00%	87.50%	87.50%	100.00%
RF00055	snoZ37	9	100.00%	100.00%	88.89%	100.00%	88.89%
RF00059	THI	118	98.31%	98.31%	77.97%	93.22%	95.76%
RF00066	U7	47	100.00%	100.00%	80.85%	95.74%	85.11%
RF00067	U15	18	100.00%	100.00%	88.89%	94.44%	88.89%
RF00080	yybP-ykoY	25	100.00%	100.00%	92.00%	88.00%	88.00%
RF00096	U8	49	100.00%	100.00%	93.88%	100.00%	100.00%
RF00101	SraC_RyeA	13	100.00%	100.00%	100.00%	100.00%	100.00%
RF00104	mir-10	11	100.00%	100.00%	72.73%	81.82%	100.00%
RF00114	S15	80	98.75%	98.75%	86.25%	95.00%	96.25%
RF00163	Hammerhead_1	75	98.67%	98.67%	82.67%	98.67%	98.67%
RF00165	Corona_pk3	14	100.00%	100.00%	100.00%	92.86%	100.00%
RF00167	**Purine**	**133**	**99.25%**	**99.25%**	76.69%	**100.00%**	**100.00%**
RF00168	Lysine	47	100.00%	100.00%	97.87%	97.87%	97.87%
RF00169	SRP_bact	468	99.15%	98.93%	85.47%	98.29%	98.07%
RF00170	msr	10	90.00%	80.00%	60.00%	70.00%	80.00%
RF00174	**Cobalamin**	**439**	**97.72%**	**97.72%**	80.41%	**98.86%**	**99.09%**
RF00177	SSU_rRNA_5	341	96.19%	97.07%	98.24%	95.89%	42.82%
RF00206	U54	22	100.00%	100.00%	95.45%	100.00%	100.00%
RF00213	snoR38	19	100.00%	100.00%	78.95%	94.74%	100.00%
RF00230	**T-box**	**65**	**90.77%**	**90.77%**	100.00%	**96.92%**	**96.92%**
RF00234	glmS	18	100.00%	100.00%	100.00%	94.44%	94.44%
RF00373	RNaseP_arch	72	100.00%	100.00%	100.00%	95.83%	93.06%
RF00379	ydaO-yuaA	106	100.00%	100.00%	90.57%	87.74%	97.17%
RF00380	ykoK	96	100.00%	100.00%	98.96%	100.00%	100.00%
RF00448	IRES_EBNA	6	100.00%	100.00%	66.67%	100.00%	100.00%
RF00504	gcvT	53	100.00%	100.00%	84.91%	92.45%	96.23%
Avg			97.51%	97.67%	89.28%	93.70%	93.73%

Table 5.2. Search results of RNAv and Infernal on RF00023/ RF00024/RF00029/RF00230 dataset.

	RF00023		RF00024		RF00029		RF00230	
	RNAv	Infernal	RNAv	Infernal	RNAv	Infernal	RNAv	Infernal
Number of Training Sequences	227	227	36	36	112	112	64	64
Filter Used	HMM	HMM/QDB	N/A	N/A	N/A	N/A	N/A	N/A
Number of NULL stem	5	N/A	5	N/A	5	N/A	5	N/A
Number of Genomes Searched	228	228	37	37	113	113	65	65
Accuracy	100%	98.25%	97.3%	94.59%	99.12%	93.81%	96.92%	90.77%

Table 5.3. Statistics of stems in RF00023 alignment

Stem Id	N	L	K	J	A	E	F	B	G	C	I	H	D	M
Good Stem	214	217	63	224	195	191	153	189	115	178	117	131	179	212
Null Stem	14	0	0	2	9	4	0	1	2	10	7	7	5	16
Weak Stem	0	11	165	2	24	33	75	38	111	40	104	90	44	0

Table 5.4. Statistics of NULL stem in the RF00023 alignment and candidate hit alignment

Stem Id	N	L	K	J	A	E	F	B	G	C	I	H	D	M	Σ
The original alignment file	14	0	0	2	9	4	0	1	2	10	7	7	5	16	77
Candidate	13	2	19	6	34	15	18	16	24	6	10	45	5	3	216
Real NULL stem in candidate	12	1	10	0	16	8	9	6	16	6	6	34	3	3	130

Table 5.5. Summary of stem and NULL stem in RF00023 alignment and candidate hit alignment

	Stem	NULL stem
The original alignment file	2976	77
Candidate	2345	30
Sensitivity	78.8%	42.86%

Table 5.6. Statistics of the stems in RF00024 alignment

Stem Id	F	E	D	C	B	A	M	L	K	J	H	G	I	Q	P	O	N
Good	32	27	22	33	37	37	37	34	37	36	37	29	37	34	35	37	35
Null Stem	4	4	13	0	0	0	0	2	0	0	0	3	0	0	0	0	0
Weak	1	6	2	4	0	0	0	1	0	1	0	5	0	3	2	0	2

Table 5.7. Statistics of the NULL stem in RF00024 alignment and candidate hit alignment

Stem Id	F	E	D	C	B	A	M	L	K	J	H	G	I	Q	P	O	N	Σ
Original sequence	4	4	13	0	0	0	0	2	0	0	0	3	0	0	0	0	0	26
Candidate hit	8	8	3	5	1	2	0	1	11	0	1	6	0	1	11	6	2	66
Real NULL stem in candidate	5	3	3	0	0	0	0	1	0	0	0	4	0	1	1	0	0	18

Table 5.8. Summary of stem and NULL stem in RF00024 alignment and candidate hit alignment

	Stem	NULL stem
The original alignment file	563	26
Candidate	271	9
Sensitivity	48.13%	34.6%

Table 5.9. Statistics of the stems in RF00029 alignment

Stem Id	B	A	C
Good Stem	112	97	74
Null Stem	0	0	0
Weak Stem	1	16	39

Table 5.10. Statistics of the NULL stem in RF00029 alignment and candidate hit alignment

Stem Id	B	A	C
Original sequence	0	0	0
Candidate hit	0	0	0
Real NULL stem in candidate	0	0	0

Table 5.11. Summary of stem and NULL stem in RF00029 alignment and candidate hit alignment

	Stem	NULL stem
Alignment	339	0
Candidate	292	0
Sensitivity	86.14%	N/A

Table 5.12. Statistics of the stems in RF00230 alignment

Stem Id	E	D	C	B	A	F	G	H	I
Good Stem	51	0	17	8	65	48	47	64	65
Null Stem	0	0	0	0	0	6	9	1	0
Weak Stem	14	65	48	57	0	11	9	0	0

Table 5.13. Statistics of the NULL stem in RF00230 alignment and candidate hit alignment

Stem Id	E	D	C	B	A	F	G	H	I	Σ
Original sequence	0	0	0	0	0	6	9	1	0	16
Candidate hit	4	2	7	0	0	5	6	7	1	32
Real NULL stem in candidate	2	2	6	0	0	4	6	1	0	21

Table 5.14. Summary of stem and NULL stem in RF00230 alignment and candidate hit alignment

	Stem	NULL stem
Alignment	553	16
Candidate	369	10
Sensitivity	66.73%	62.5%

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

This dissertation studies the problem of fast and accurate searching for non-coding RNA pseudoknot structures in genomes. Based on the conformational graph model for RNA pseudoknots, given k candidate regions in the target sequence for each of the n stems in the structure, we could compute a best alignment in time $O(k^{t+1}n)$ based upon a tree width t decomposition of the structure graph. Through search tests on the implemented program RNATOPS [1], we have shown its performance comparable with or better than that of Infernal [2] and FastR [3] in identifying large or complex RNA structures including pseudoknots; Also, this dissertations presents results of an investigation toward developing a new framework for distant ncRNA structural homolog search. In this work, secondary structure conformations are modeled as graphs with small tree width and sequence-structure alignment for homolog detection is formulated as graph homomorphism. The technique based on the notion of NULL stem is used to resolve the issue of optional stems that may be deleted from the structure profile or may be due to a misalignment. Test results on 51 benchmark data sets of Infernal (9 of them containing pseudoknots) show that a program based on these methods, RNAv, with the capability of detecting pseudoknots, has a comparable performance to the latest version of Infernal, and is better in detection of some distant homologs.

6.2 FUTURE WORK

Following the footstep from RNATOPS, a ncRNA secondary structure prediction tool, and RNAv, a ncRNA structure variation search tool, we consider the following two future developments.

6.2.1 PROBABILISTIC GRAPH MODEL FOR RNA STRUCTURE EVOLUTION

Investigations on evolution of RNA secondary structure have just begun recently, mainly for this interest of profiling structure for search. In previous work [4-8] RNA base and base pair evolution information was incorporated into SCFG models. An improved model for RNA structural evolution based on a transducer composition algorithm has also been proposed [9, 10] which can deal with limited degree of structural variations but has yet to be incorporated into a search program. The program, trCYK[11], a local alignment algorithm for Infernal, contains a technical solution that addresses the issue of aligning the structural model with incomplete sequences. The scoring is based on conserved primary sequence and structure information, instead of a structural evolution model. To date, a general method that addresses both possible misalignments and structural variation is still missing [12]. Searches for structurally distant homologs still largely rely on customized methods or tools [13].

To incorporate substantial structural variations, we propose to extend the conformational graph to be the probabilistic conformation graph model. Formally, this is to assign a probability to every edge (i.e., every non-directed edge). Recall that each non-directed edge models a stem and each directed edge models a loop. Each directed edge automatically gets the probability 1 while each non-directed edge, e, will be assigned a probability $P(e=1)=r, 0<r\leq 1$. This probability is for the corresponding stem to be present in the structure profile, observed from the

available sequences in the family. On the other hand, $p(e=0)=1-r$ is the probability for the stem to be absent.

A probabilistic non-directed edge, e, defined in the conformational graph specifies the chance for the corresponding stem to be present and absent. One can think of this in terms of stem insertion/deletion when a target sequence is aligned to the structure profile modeled with the conformational graph. To see this, assume $p(e=1)\le 1/2$. If the optimal alignment manifests the presence of the stem, it is a stem insertion. Conversely, it can be thought of a stem deletion for the case when $p(e=0)\ge 1/2$ and the stem is absent from the target sequence. The other two cases are just normal matches.

Though dependent on outcomes from the ncRNA structural evolution study, we believe it will also be necessary to introduce joint and conditional probabilities for edges in the conformational graph in order to account structural variations more accurately. For instance, to allow insertions and deletions of a substructure, e.g., consisting of two side-by-side stems e_1 and e_2, in the profile, the joint probability $p(e_1,e_2)$ and conditional probabilities $p(e_1|e_2)$ and $p(e_2|e_1)$ will become handy. For example, Figure 6.1 shows the joint probability $p(e_1,e_2)=p(x_1,y_1)\times p(x_2,y_2)$. By default, stems are independent unless otherwise defined. More complex substructures will be likewise included.

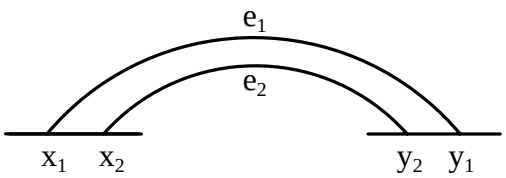


Figure 6.1. The joint probability $p(e_1,e_2)=p(x_1,y_1)\times p(x_2,y_2)$

Therefore, the extended conformational graph defines well an ncRNA secondary structure profile, a probabilistic space of secondary structures with a set of random variables that are edges (of discrete values 0/1), each of which corresponds to a unique stem in the profile. The probability function P, for various random variables will be determined from the structural evolution study.

6.2.2 NUCLEOTIDE-LEVEL TOPOLOGY GRAPH MODELING

The graph model can easily profile structures caused by nucleotide interactions beyond the base pairing. For example, the graph model makes it easy to profile tertiary interactions or triple helices recently found in the telomerase RNA genes of human and yeast genomes [14-17]. Although one of the two stems involved in such a triple helix is actually formed by two base pairing regions that are arranged in the same direction (5′ to 3′), our approach will allow the stem to be modeled with an individual CM the same way as modeling a regular stem, without the need of additional, new techniques.

But there may be one problem here. In our current framework, either RNATOPS [1] or RNAv [18], stem and loop models are introduced from the profile training data, to some degree forces the program to look for the stem in its corresponding sequence region even though there are no actual stem there, or those region actually contains one arm of stem which has interaction with the other arm in other distant sequence region. The latter case actually is what researcher is working on to search the protein 3D or RNA 3D structure. Because this limitation exists in RNAv, it would be difficult to apply RNAv directly to solve the protein 3D or RNA 3D structure prediction problem. Here we are thinking about removing the stem and loop models from the profile training data and, in the conformational graph, changing *arc* to be the interaction of canonical base pairing. Then this graph model, on the nucleotide level, will become more

flexible to incorporate any interaction with more distant sequence regions while the tree-width remains small (Figure 6.2) for graphs at nucleotide level (the average tree-width is 3.16) [19].

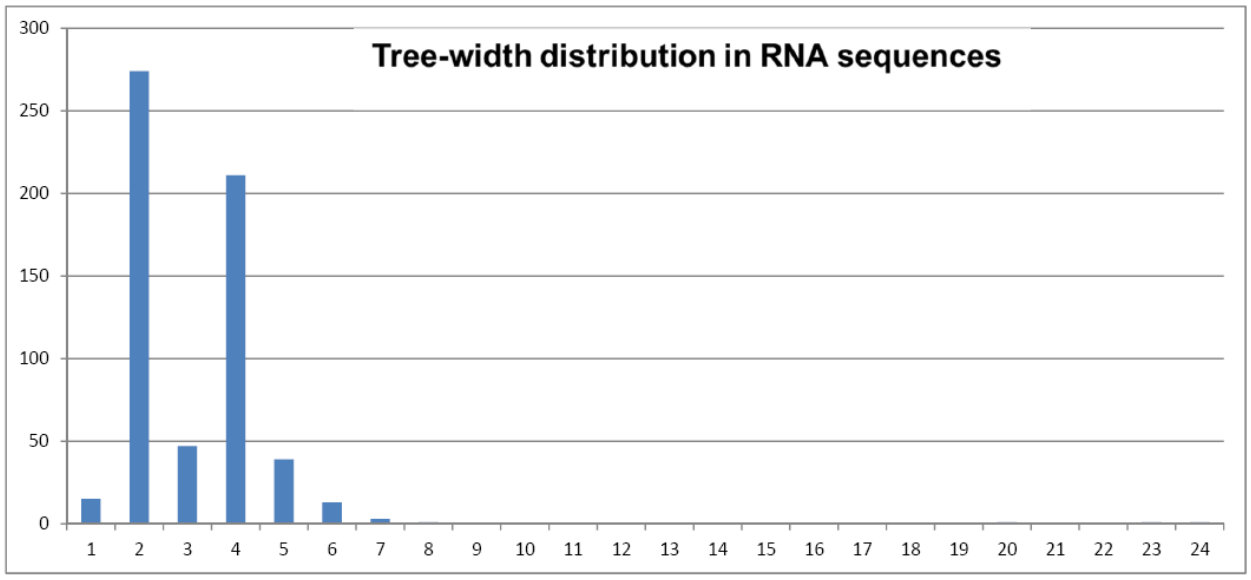


Figure 6.2 Treewidth distribution on the nucleotide level [19]

6.3 SUMMARY OF THE CHAPTER

This chapter summarized the previous chapters of this dissertation. Searching genomes for ncRNAs by their secondary structure is an important goal for bioinformatics. This dissertation provides fast and accurate search method for RNA pseudoknot structures, also structure variation search for ncRNA secondary structures. This dissertation provides an original perspective on this structure variation problem and shows that it can be modeled to be the GRAPH HOMOMORPHISM problem.

For the future work, we extend the graph model to be the probabilistic graph model for the RNA structure evolution problem and nucleotide-level topology graph model for the Protein 3D or RNA 3D structure prediction problem. We are glad to see our work actually provides a general framework for structure search in bioinformatics. To solve different problems, we need

more specific graph model to fit into the problem domain, and finding answers to them would be an interesting extension of the work presented in this dissertation.

6.4 REFERENCES

[1]. Huang, Z., Y. Wu, J. Robertson, L. Feng, R. Malmberg, and L. Cai. 2008. Fast and accurate search for non-coding rna pseudoknot structures in genomes. Bioinformatics. 24:2281–2287.

[2]. Eric P. Nawrocki, Diana L. Kolbe, and Sean R. Eddy, Infernal 1.0: inference of RNA alignments, Bioinformatics. 2009 May 15;25(10):1335-7.

[3]. Zhang, S., B. Haas, E. Eskin, and V. Bafna. 2005. Searching genomes for noncoding RNA using FastR. IEEE/ACM Trans. Comput. Biol. Bioinform. 2:366–379.

[4]. Knudsen B, Hein J (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics 15: 446–454.

[5]. Knudsen B, Hein J (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. Nucleic Acids Res 31: 3423–3428.

[6]. Pedersen JS, Meyer IM, Forsberg R, Simmonds P, Hein J (2004a) A comparative method for finding and folding RNA secondary structures within protein-coding regions. Nucleic Acids Res 32: 4925–4936.

[7]. Pedersen JS, Forsberg R, Meyer IM, Hein J (2004b) An evolutionary model for protein-coding regions with conserved RNA structure. Mol Biol Evol 21: 1913–1922.

[8]. Pedersen, J.S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E., Rogers, J., Kent, J., Miller, W., and Haussler, D. Identification and Classification of Conserved RNA Secondary Structures in the Human Genome. PLoS Computat Biol 2(4), e33 (2006).

[9]. Bradley RK, Holmes I (2009) Evolutionary Triplet Models of Structured RNA. PLoS Comput Biol 5(8): e1000483. doi:10.1371/journal.pcbi.1000483.

[10]. Holmes I: A probabilistic model for the evolution of RNA structure. BMC Bioinformatics 2004, 5:166.

[11]. Kolbe DL, Eddy SR, Local RNA structure alignment with incomplete sequence, Bioinformatics, 25(10):1236-1243, 2009.

[12]. Andreas R. Gruber, et al. 2010. Rnaz 2.0: Improved Noncoding Rna Detection, Pacific Symposium on Biocomputing 15:69-79.

[13]. Axel Mosig. et al. (2009) Customized strategies for discovering distant ncRNA homologs, Briefings in Functional Genomics and Proteomics, doi:10.1093/bfgp/elp035.

[14]. Chen,L. and Greider,C.W. (2004)An emerging consensus for telomerase RNAstructure. Proc. Natl Acad. Sci. USA, 101, 14683–14684.

[15]. Lin,J. et al. (2004) A universal telomerase RNA core structure including structured motifs required for binding the telomerase reverse transcriptase protein. Proc. Natl Acad. Sci. USA, 101, 14713–14718.

[16]. Shefer,K. et al. (2007) A triple helix within a pseudoknot is a conserved and essential element of telomerase RNA Mol. Cell Biol., 27, 2130–2143.

[17]. Theimer,C.A. et al. (2005) Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. Mol. Cell, 17, 671–682.

[18]. Huang,Z., Malmberg,R., Mohebbi, M and Cai,L. 2010. RNAv: Non-coding RNA Secondary Structure Variation Search via Graph Homomorphism, In Proceedings of Computational Systems Bioinformatics Conference (CSB 2010), August, 2010. Vol. 9, p. 56-69.

[19] Mohebbi M and Cai,L. unpublished result. 2011.