MBLAST-PK: A FAST ALGORITHM FOR RNA PSEUDOKNOTTED STRUCTURE PREDICTION

by

JIANLIANG DAI

(Under the Direction of Liming Cai)

ABSTRACT

RNA secondary structure including pseudoknots is computationally difficult to predict. Almost all existing algorithms for optimal pseudoknot prediction entail $O(n^6)$ running time and $O(n^4)$ memory space even for restricted categories of pseudoknots, making pseudoknot prediction unrealistic for RNA of more than 100 nucleotides.

We introduced a new heuristic algorithm mBLAST-PK for RNA pseudoknot prediction that can substantially reduce the computational costs. The new algorithm preprocesses the RNA sequence to find all base pairing regions in a sequence based on a modified BLAST technique. It then non-trivially extends Nussinov folding to calculate the structure including pseudoknots. Our program predicted the RNA pseudoknot structures in bacterial tmRNA sequences at about 81% accuracy. The running time and memory space consumption by the algorithm are both reduced by at least two orders of magnitude, making the task of pseudoknot prediction routine on desktop computers.

INDEX WORDS: RNA, Pseudoknot, BLAST, Nussinov algorithm, dynamic programming

MBLAST-PK: A FAST ALGORITHM FOR
RNA PSEUDOKNOTTED STRUCTURE PREDICTION

by

JIANLIANG DAI

B.S., Zhejiang Forestry College, P.R. China, 1993

M.S., Beijing Forestry University, P.R. China, 1996

PHD, The University of Georgia, 2003

A Thesis Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of the Requirement for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2004

MBLAST-PK: A FAST ALGORITHM FOR

RNA PSEUDOKNOTTED STRUCTURE PREDICTION

by

JIANLIANG DAI

Major professor:	Liming Cai
Committee:	Russell Malmberg Eileen Kraemer

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2004

To my beloved parents and wife

ACKNOWLEDGEMENTS

I would like to express sincere thanks to my major professor, Dr. Liming Cai, for his invaluable advice, honest criticism, endless patience, and most importantly, his trust and encouragement throughout my degree program. I would like to thank Dr. Russell Malmberg and Dr. Eileen Kraemer for their kind advice, helpful comments and friendly criticism.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

Ribonucleic acid (RNA) is an important macromolecule in most living organisms. RNA molecules play essential roles in many processes such as translation and gene regulation. Single stranded RNA molecules have the property that they can fold to form intramolecular base-paired structures (double helices), either in a nested and parallel fashion called stem-loops or in a crossing fashion called pseudoknots. These structures are essential to the function of RNA molecules. Pseudoknots were found in many important RNAs, with a role in translation [8], viral genome structure [14] and ribozyme active sites [20]. Computational prediction of RNA structure can be made for two different types of sequence data: either a single RNA sequence or multiple RNA sequences.

Due to its inherent complexity, an RNA pseudoknot structure is computationally difficult to predict [6]. Algorithms currently available for optimal pseudoknot prediction are evolved from stem-loop prediction algorithms based on thermodynamics, formal grammars, graph theoretics, and other theoretical frame works.

For single RNA sequences, the Nussinov dynamic programming algorithm finds the stem-loop structure by computing the maximum number of base pairings [13, 6]. The algorithm checks four ways in which the best structure for a subsequence X[i..j] from position i to position j can be made from the best structure for smaller subsequences by adding the nucleotide at position i and/or the one at position j: (a) Add an unpaired base i to the best structure for the subsequence X[i+1, j]; (b) Add an unpaired base j to the best structure for the subsequence X[i, j-1]; (c) Add paired bases i-j to the best structure for the subsequence X[i+1, j-1]; (d) Combine two optimal substructures i, k and k+1, j (Figure 1-1). Combined with thermodynamic parameters, the Nussinov RNA folding algorithm was later developed into widely used software systems such as MFOLD and Vienna [23, 24]. These programs can predict non-pseudoknot structure for any given single RNA sequence at 70% accuracy. Recently thermodynamic energy minimization is extended to find pseudoknots[12]. In particular, Rivas and Eddy created a dynamic programming algorithm based on MFOLD which can find optimal folding in O(n^6) time and O(n^4) space [15]. There are other algorithms developed for pseudoknot prediction based on thermodynamics, however these methods do not guarantee optimality [1, 12].

```
     o         o              o             o          o
   o   o     o   o          o   o         o   o      o   o
    o-o       o-o            o-o           o-o        o-o
    o-o       o-o            o-o           o-o        o-o
i+1 o-o j   i o-o j-1    i+1 o-o j-1    i o-o--o--o--o-o j
 i o              o j       i o-o j              k  k+1
```

(a) i unpaired (b) j unpaired (c) i,j pair (d) bifurcation

a. Add an unpaired base *i* to the best structure for the subsequence *i*+1, *j*

b. Add an unpaired base *j* to the best structure for the subsequence *i*, *j*-1

c. Add paired bases *i*-*j* to the best structure for the subsequence *i*+1, *j*-1

d. Combine two optimal substructures *i*, *k* and *k*+1, *j*

Nussinov RNA folding algorithm

Initialization:

$S\ (i, i) = 0$ for $i = 1$ to L,

$S\ (i, i\text{-}1) = 0$ for $i = 2$ to L,

Recursion:

$$S(i, j) = \max \begin{cases} S\ (i+1, j), & \text{(a)} \\ S\ (i, j-1), & \text{(b)} \\ S\ (i+1, j-1) + \delta(i, j), & \text{(c)} \\ \max_{i<k<j} [S\ (i, k) + S(k+1, j)]. & \text{(d)} \end{cases}$$

Fig. 1-1. The Nussinov RNA folding algorithm finds the structure with the most base pairs. The algorithm checks four ways in which the best structure for subsequence from position i to position j can be made from the best structure for smaller subsequences by adding the nucleotide at position i and/or the one at position j. Picture modified from http://ludwig-sun2.unil.ch/~bsondere/nussinov/.

Comparative sequence analysis is considered to be one of the most reliable means of detecting an RNA secondary structure [6]. Pseudoknots were predicted in flavivirus RNA using comparative sequence analysis [25]. However, this method requires a structurally correct multiple alignment of RNAs for covariance analysis. If only a single sequence or a small family of RNAs with little sequence diversities is available, this method cannot be applied [15].

Some heuristic approaches have been introduced for RNA single-sequence structure prediction. For example, a quasi-Monte Carlo search method and several genetic algorithms exist for pseudoknot prediction, but these methods are not guaranteed to find the optimal structures [18, 26, 27]. Recently, Dirks and Pierce described an $O(n^5)$ dynamic programming for computing the partition function and minimum energy structure over a pseudoknot. However, the testing results are subject to the limitations of the approximate physical model on which the partition function is based [29].

A different approach to pseudoknot prediction is based on various aspects of graph theory. A tree adjoining grammar was used for RNA pseudoknot prediction [22], but later it was found that the tree adjoining grammar is not critical while the parsing procedure is important [1]. Another method is called Maximum Weighted Matching (MWM) [4, 19]. In MWM, the graph represents bases as vertices and all possible interactions as weighted edges among all the bases (vertices) for the RNA. MWM gives the RNA for folding with matching having the maximal summed edge weights. However, MWM seems to work best to folding sequences for which a previous multiple alignment exists [15]. MWM also has a problem of producing spurious base pairs [4].

Alternatively, when the consensus structure of a set of aligned homologous RNA sequences is available through comparative sequence analysis, a covariance model can be constructed for the structure using stochastic context-free grammars (SCFG). With a SCFG probabilistic model, not only can single-sequence structure prediction be performed, but also profiles can be developed for structural homology recognition in database searches. Although RNA stem-loops have been successfully modeled with SCFG, RNA pseudoknot modeling requires context-sensitive grammars, which are much more complex and clumsy to implement. As a result, most efforts have been made to model pseudoknots with the other kinds of grammars, essentially extensions of SCFG [15]. For example, Cai *et al* [2003] introduced a grammar modeling approach for pseduoknot structure based on parallel communicating grammar systems (PCGS) and showed that technically a pseudoknot model specification could be as simple as SCFG [5]. But most of these grammars are still very complicated [16, 22] and computationally intensive as well as memory consuming [7].

For consensus pseudoknot structure prediction, there are a small number of algorithms that are based on thermodynamic methods [17, 9], genetic algorithms [10] and graph theory [11]. These methods while successful at their specific aims, are computationally resource consuming in addition to being lack in generality.

Algorithms currently available for optimal pseudoknot prediction requires $O(n^4)$ memory and $O(n^6)$ run-time even for restricted pseudoknot categories. The memory and CPU time requirements of these algorithms have made it impossible to predict pseudoknots for an RNA sequence longer than 200 nucleotides [21]. Improvements in space and time complexity are needed. Generally speaking, RNA structure prediction could be speeded up either through implementation techniques such as parallel computing or through the use of heuristic methods with some tradeoffs.

In this study we propose a novel approach to speed up and reduce the memory cost of single-sequence RNA pseudoknot prediction. The proposed algorithm extends and combines the idea of BLAST and Nussinov folding algorithms. The BLAST algorithm was developed to find local alignments between a query sequence and a target database [2]. It is based on the idea that high score alignments are very likely to contain a short stretch of exact matches. The BLAST algorithm first creates a list of scoring words of length W, then uses them as seeds to perform database search and identify exact matches (hits). For each word match, BLAST extends the alignment in both direction to find alignments that score greater than the threshold (Figure 1-2). Since there are similarities between the RNA folding process and sequence local alignment, pre-proccessing the RNA sequence to find all the base pairing regions that can form double helices may provide a solution to speeding up RNA structure prediction, especially for pseudoknots. In particular, the Nussinov folding algorithm can be modified to work on the base pairing regions rather than on individual bases. For pseudoknots, the Nussinov algorithm is extended based on a similar idea developed for stochastic grammar based prediction [5]. It is expected as well that the method may provide feasible solutions to multiple-sequence consensus structure prediction.

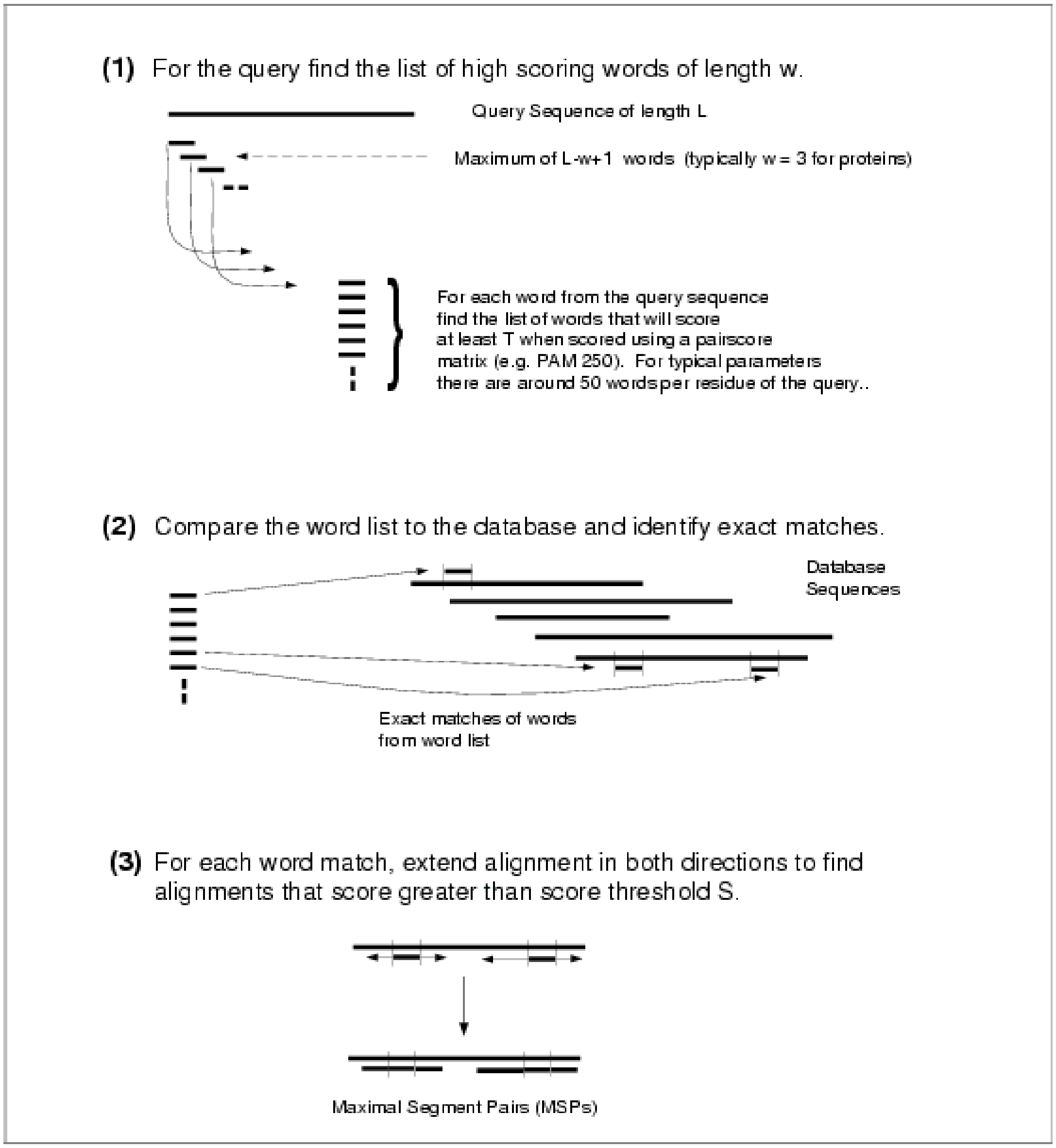

(1) For the query find the list of high scoring words of length w.
Query Sequence of length L
Maximum of L-w+1 words (typically w = 3 for proteins)
For each word from the query sequence find the list of words that will score at least T when scored using a pairscore matrix (e.g. PAM 250). For typical parameters there are around 50 words per residue of the query..
(2) Compare the word list to the database and identify exact matches.
Database Sequences
Exact matches of words from word list
(3) For each word match, extend alignment in both directions to find alignments that score greater than score threshold S.
Maximal Segment Pairs (MSPs)

Fig. 1-2. The BLAST algorithm works in three steps: preproccessing of the query, exact-match database search, and extensions of the hits. Figure taken from [30].

CHAPTER 2

MBLAST ALGORITHM

We propose a novel algorithm for efficient RNA pseudoknot prediction .The algorithm, mBLAST-PK, consists of two major components. The function of the first component is finding all the existing base-pairing regions in the RNA sequence. Each of these base-pairing regions can form a double helix with some other base-pairing region. It is accomplished with a modified BLAST algorithm.

The original BLAST algorithm involves three distinct steps: preproccessing of the query, exact-match database search and extensions of the hits [30]. Instead of doing an exhaustive database search, this algorithm pre-processes the query sequence based on a list of seeds and performs heuristic seed-search, thereby reducing computation time dramatically. The RNA folding process could be thought of as a local alignment process, in which one short base region aligns to another complementarily pairing base region (to form a double helix). Following this idea, we modified the BLAST algorithm so that it could be used to find all the base-pairing regions that may form double helices. The modified BLAST algorithm (Figure 2-1) proceeds in three steps: generation of a complementary sequence, exact-match search, and extension of the hits.

(1) Generation of a complementary sequence: RNA structure consists of a number of double helices. Each double helix is formed by two base-pairing regions. A new RNA sequence is first generated that is complementary to the original RNA sequence. Assume that a complementary sequence X' was generated from the sequence X of length N. The nucleotide of X' at position k is the complementary base of the nucleotide at position N-k+1 of X.

(2) Exact-match search: Starting from the 5' end of sequence X', for each position k, every 3 or 4 consecutive nucleotides from 5' to 3' are used as a seed (assume the minimum length of a double helix is 3 or 4), to scan through the original RNA sequence from 5' to 3'. That is, we compare every 3 or 4 consecutive nucleotides in X' from 5' to 3' with every 3 or 4 consecutive nucleotides in X from 5' to 3'. We also use 4 consecutive nucleotides in X' from 5' to 3' as a seed when one of 4 nucleotides is C or A (correspondent to G or U in X). If the compared nucleotide in X is U or G, it is also a hit, so that the non-canonical pair G-U is allowed in the stacked base pairs, which occurs quite often. We stop scanning when the distance between the

last position compared in X and the first position of the seed in X (N-1-k-2) is less than the predefined minimum loop length. Once we get a hit (exact match), we perform step (3).

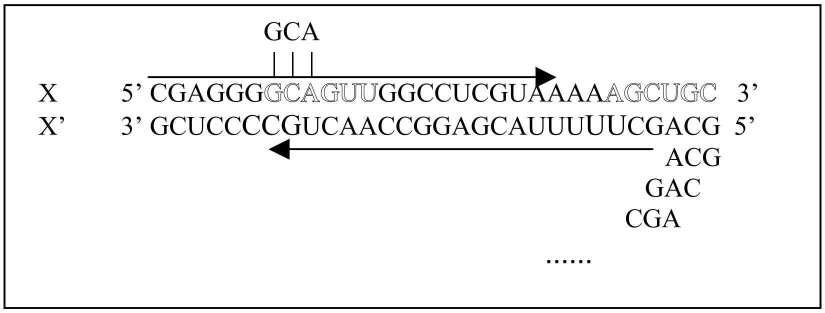

GCA
X 5’ CGAGGGGCAGUUGGCCUCGUAAAAAGCUGC 3’
X’ 3’ GCUCCCCGUCAACCGGAGCAUUUUUCGACG 5’
ACG
GAC
CGA
......

Fig. 2-1. The mBLAST algorithm scans the RNA sequence X using a seed GCA from the complementary sequence X’, identify an exact-match (hit) and extend the hit to find two longer base pairing regions (outlined) Ungapped extensions are performed.

(3) Extension of the hits: once we get a hit, we perform ungapped extension in the direction of 5’ to 3’ for both sequence X’ and sequence X by continuing to compare the nucleotides at the following positions from X and X’. Mismatch extension (One or two mismatches) is allowed only under the condition that mismatches are followed by more matches so that the net log odds score for the mismatch extension is positive. The non-canonical G-U pair is also allowed during the ungapped extension.

This algorithm runs in time O (N^2). The types of base regions that could be identified using this algorithm depend on the types of matches searched in step (2) and/or in step (3). For example, if a match consists of 5 matched base pairs, then the regions identified will have at least 5 base pairs. If one mismatch is allowed in the middle of comparisons in step (2) and/or in step (3), then the double helix formed by these two base regions will include a pair of bulges. If such information is available about the RNA structure and incorporated into step (2) and/or step (3), the accuracy of structure prediction could be greatly enhanced.

A hit in step (2) followed by extensions in step (3) identifies two base-pairing regions in the original sequence. The region from sequence X is represented by symbol α(i) (which essentially is the first position index i of this base region), and the other pairing region in X is represented by symbol β(j) (which essentially is the last position index j of this base region, i.e. the first position of the seed k in X' and N-k+1 in X). Note that the index α(i) or β(j) may refer to more than one base region, because there may be more than one base region starting at this position but with the different length. Therefore, a single α(i) or single β(j) cannot uniquely identify a double helix, only α(i) together with β(j) can uniquely determine a double helix. At this point, the sequence X could be represented by "*a base region sequence*", namely, a sequence of base regions indexed by α(i) and β(j).

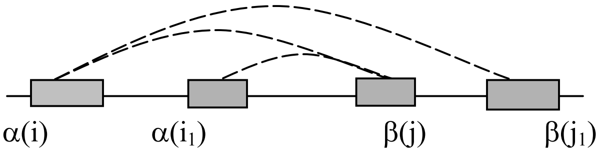


Fig. 2-2. *A base region sequence*; One base region may have multiple complementary base regions.

▭ α(i) a base region starting at the position i1, ▭ β(j) a base region ending at the positon j ,

——— the RNA backbone ▭⌒▭ two base pairing regions are considered in the current optimal solution calculation.

The log odds scores are calculated for every complete double helix formed by α(i) and β(j) and its three subhelices. The log odds score of a double helix is the summation of the log odds ratio of every base pair $S = \sum_i s(x_i, y_i)$, where $s(a,b) = \log\left(\frac{p_{ab}}{q_a q_b}\right)$ is the log odds ratio of a base pair (a, b)[6]. The base probability q_a and q_b, and the base-pairing probability p_{ab} are obtained from the training data set described before [5].

The purpose of considering subhelices is that although the complete helix is excluded because it overlaps with other double helices, one of these subhelices still could be included in the prediction result. These three sub-helices share the same starting positions i and j, but with different lengths (i.e. for a double helix of length L, then three sub-helices have the lengths of L-1, L-2 and L-3 respectively. The mismatch extension part is treated as a single residue). The log-odds scores and their associated lengths are stored in tables. If the region α(i) does not complementarily pair with region β(j), all these values are set to zero. If any of L-1, L-2 and L-3 is less than the assumed minimal length of a double helix, the relevant value(s) are set to zero.

CHAPTER 3

NUSSINOV-PK ALGORITHM

The second component of the mBlast-PK algorithm is in the spirit of Nussinov folding algorithm. The Nussinov algorithm was modified and extended so that it treats each base-pairing region as a single nucleotide and is capable of finding all the base regions that could form a structure, including pseduoknots, with the highest score.

The Nussinov algorithm component calculates the structure with the maximum number of base-pairs. The dynamic programming recursively calculates the best structures for smaller subsequences and works it ways outwards to larger subsequences [6]. I modified Nussinov algorithm to handle the indices of base regions $\alpha(i)$ and $\beta(j)$ rather than the nucleotide indices i and j. Two complementary base regions form one double helix in a RNA sequence. Formally, let $\alpha(i)$ represent the base region that is close to the 5' end of the RNA sequence and this region starts at position i, $\beta(j)$ refers to the complementary base region that is close to the 3' end of the RNA sequence and this region ends at position j. Let function first($\alpha(i)$) return i, the first position of base region $\alpha(i)$, function last($\beta(j)$) return j, the last position of base region $\beta(j)$. Let $X[\alpha(i)..\beta(j)]$ be the subsequence from position i to position j. We say $\alpha(i)$ is complementarily matchable (simply, matchable) with $\beta(j)$ when they can form a stable double helix. Symbol $h(\alpha(i), \beta(j))$ indicates the double helix formed by base region $\alpha(i)$ and base region $\beta(j)$. Note that j must be larger than or equal to i+8 for subsequence $X[\alpha(i)..\beta(j)]$ to form a stem-loop structure if we assume that a double helix has at least 3 base pairs and the minimum loop length is 3. Let $S(\alpha(i), \beta(j))$ be the maximum score for $h[\alpha(i)..\beta(j)]$. Let $l(\alpha(i), \beta(j))$ be the number of base pairs and $\delta(\alpha(i), \beta(j))$ be the log odds score of the complete double helix $h(\alpha(i),\beta(j))$. The other important functions and indices are defined as follows:

1. next($\alpha(i)$) returns the base region starting at the first position > first($\alpha(i)$), it may overlap with the base region $\alpha(i)$,
2. prev($\beta(j)$) returns the base region ending at the first position < last($\beta(j)$), it may overlap with the base region $\beta(j)$,
3. nextNo($\alpha(i)$) returns the base region starting at the first position > last($\alpha(i)$), it cannot overlap with

the base region $\alpha(i)$,

4. $\text{prevNo}(\beta(j))$ returns the base regions ending at the first position < $\text{first}(\beta(j))$, it cannot overlap with the base region $\beta(j)$,

7. $\text{nextStart}(\beta(k))$ returns the base regions starting at the first position > $\text{last}(\beta(k))$,

Table 3-1. The extended Nussinov algorithm for calculating the score of subsequence $X[\alpha(i)..\beta(j)]$. Formula (1), (2), (3) and (4) are analogues to formula (a), (b), (c) and (d) in the Table 1-1. Formula (5) looks for a pseudoknot formed by a double helix $h(\alpha(i), \beta(j_1))$ and a second double helix $h(\alpha(i_1), \beta(j_2))$. In this case $X[\alpha(i)..\beta(j)]$ is partitioned into two parts and calculated by function $P(\alpha(i),\beta(j_1),\alpha(i_1))$ and $Q(\text{nextStart}(\beta(j_1)), \beta(j), \beta(j_2))$, respectively. These two functions are defined in table 3-2.

$$S(\alpha(i), \beta(j)) = \max \begin{cases} S(\text{next}(\alpha(i)), \beta(j)), & (1) \\ S(\alpha(i), \text{prev}(\beta(j))), & (2) \\ S(\text{nextNo}(\alpha(i)), \text{prevNo}(\beta(j))) + \delta(\alpha(i), \beta(j)), & (3) \\ \max_{i<k<j} [S(\alpha(i), \beta(k)) + S(\text{nextStart}(\beta(k)), \beta(j))], & (4) \\ \max_{i<i1<j1<j2<=j} [P(\alpha(i), \beta(j_1), \alpha(i_1)) + Q(\text{nextStart}(\beta(j_1)), \beta(j), \beta(j_2)) + & (5) \\ \qquad \delta(\alpha(i_1), \beta(j_2))]. & \end{cases}$$

The modified Nussinov algorithm is shown in the Table 3-1. Formula (1), (2), (3) and (4) in the Table 3-1 are analogues to formulas (a), (b), (c) and (d) in the Table 1-1, which shows the original Nussinov algorithm. Similarly, formulas (1), (2), (3) and (4) check the four possible ways in which the best base-paired structure for $X[\alpha(i)..\beta(j)]$ can be made from smaller subsequences. Specifically, formula (1) excludes the non-matching base region $\alpha(i)$, the best structure for subsequence $X[\alpha(i)..\beta(j)]$ that includes the same set of base regions as that for $X[\text{next}(\alpha(i))..\beta(j)]$. Formula (2) excludes a non-matching base region β, the best base-paired structure for $X[\alpha(i)..\beta(j)]$ that includes the same set of base regions as that for $X[\alpha(i)..\text{prev}(\beta(j))]$. Formula (3) adds matchable base regions $\alpha(i)$ and $\beta(j)$ to the best base-paired structure for the subsequence $X[\text{nextNo}(\alpha(i))..\text{prevNo}(\beta(j))]$; Formula (4) combines two optimal substructures for $X[\alpha(i)..\beta(k)]$ and $X[\text{nextStart}(\beta(k))..\beta(j)]$, choosing the best base region between $\alpha(i)$ and $\beta(j)$.

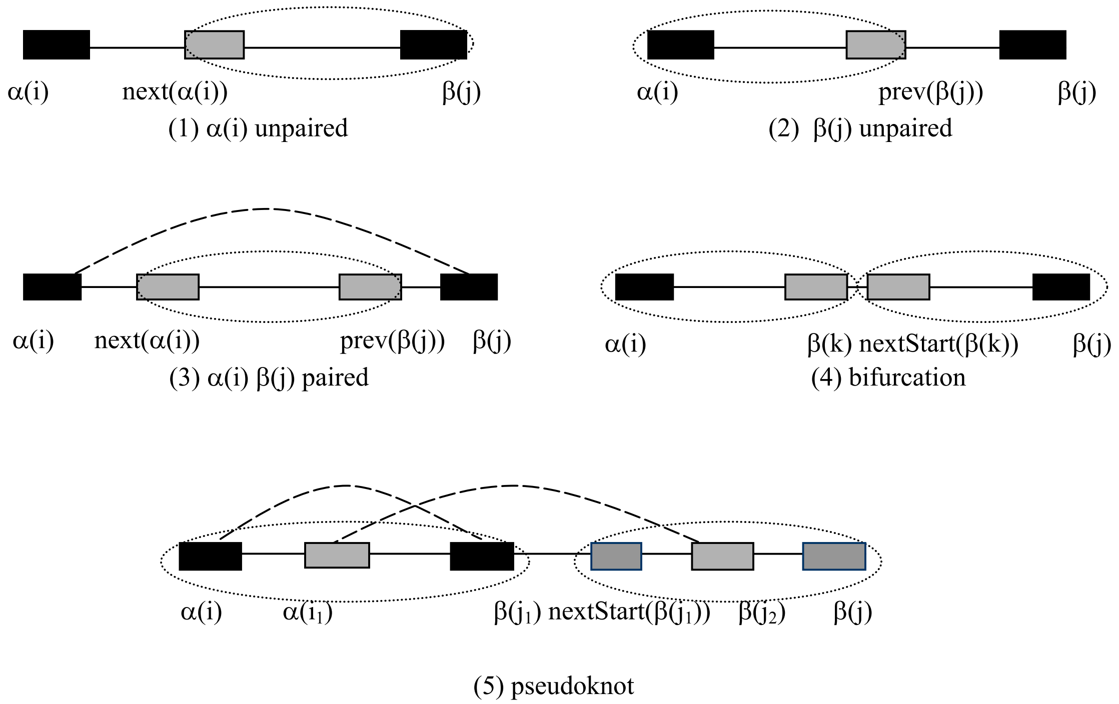


Fig. 3-1. For subsequence $X[\alpha(i)..\beta(j)]$, formulas (1), (2), (3) and (4) check four possible ways in which the best base-paired structure for $X[\alpha(i)..\beta(j)]$ can be made from smaller subsequences. Formula (5) looks for a pseudoknot formed by a double helix $h(\alpha(i), \beta(j_1))$ and a second double helix $h(\alpha(i_1), \beta(j_2))$. $X[\alpha(i)..\beta(j)]$ is partitioned into two parts (indicated by two circles): $X[\alpha(i).. \beta(j_1)]$ including region $\alpha(i_1)$, calculated by $P(\alpha(i),\beta(j_1),\alpha(i_1))$, and $X[nextStart(\beta(j_1)), \beta(j)]$ including region $\beta(j_2)$, calculated by $Q(nextStart(\beta(j_1)), \beta(j), \beta(j_2))$. Base regions $\alpha(i)$, $\beta(j_1)$, $\alpha(i_1)$ and $\beta(j_2)$ cannot be overlapped with other base regions.

In addition to checking four ways in which the best structure for $X[\alpha(i)..\beta(j)]$ can be made from smaller subsequences, we also calculate the maximum score for the pseudoknot structure possibly formed by $X[\alpha(i)..\beta(j)]$ with the formula (5). In this case, base region $\alpha(i)$ and $\beta(j_1)$ must form a double helix, and base region $\alpha(i_1)$ and $\beta(j_2)$ must form a double helix, so that double $h(\alpha(i),\beta(j_1))$ and $h(\alpha(i_1),\beta(j_2))$ form a pseudoknot (Figure 3-2). In the calculations of formula (3) and formula (5), whenever matchable base regions $\alpha(i)$ and $\beta(j)$ are considered, the complete $h(\alpha(i),\beta(j))$ and its three sub-helices are tried, the one that results in the structure with highest log odds score is used.

To calculate the pseudoknot score, $X[\alpha(i)..\beta(j)]$ is partitioned into two parts: $X[\alpha(i)..\ \beta(j_1)]$ including region $\alpha(i_1)$, calculated by $P(\alpha(i), \beta(j_1), \alpha(i_1))$, and $X[\text{nextStart}(\beta(j_1), \beta]$ including region $\beta(j_2)$, calculated by $Q(\text{nextStart}(\beta(j_1)), \beta(j), \beta(j_2))$. These two parts are not the same as, but are similar to, the "P-structure" described in Cai *et al*. [2003]. $P(\alpha(i), \beta(j), \alpha(i_1))$ and $Q(\alpha(i), \beta(j), \beta(j_1))$ are calculated by $PK(\alpha(i), \beta(j), s1, s2)$ and $QK(\alpha(i), \beta(j), s1, s2)$, respectively, which are defined in Table 3-2.

Table 3-2. The modified Nussinov algorithm for calculating $PK(\alpha(i), \beta(j), s1, s2)$and $QK(\alpha(i), \beta(j), s1, s2)$.

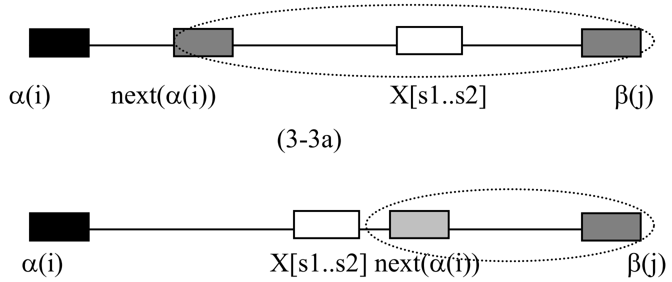
$$P(\alpha(i), \beta(j), \alpha(i_1)) = PK(\alpha(i), \beta(j), s1, s2) = QK(\text{nextNo}(\alpha(i)), \text{prevNo}(\beta(j)), s1, s2) + \delta(\alpha(i), \beta(j)). \qquad (6)$$

$$Q(\alpha(i), \beta(j), \beta(j_1)) = QK(\alpha(i), \beta(j), s1, s2). \qquad (7)$$

$$QK(\alpha(i),\beta(j),s1,s2)=\max\begin{cases}
QK(\text{next}(\alpha(i)), \beta(j), s1, s2), & \text{if } (\text{last}(\text{next}(\alpha(i)))<s1) \qquad (8a)\\
S(\text{next}(\alpha(i)), \beta(j)), & \text{if } (s2<\text{first}(\text{next}(\alpha(i)))) \qquad (8b)\\
QK(\alpha(i), \text{prev}(\beta(j)), s1, s2), & \text{if } (s2<\text{first}(\text{prev}(\beta(j)))) \qquad (9a)\\
S(\alpha(i), \text{prev}(\beta(j))), & \text{if } (\text{last}(\text{prev}(\beta(j)))<s1) \qquad (9b)\\
QK(\text{nextNo}(\alpha(i)), \text{prevNo}(\beta(j)),s1,s2)+\delta(\alpha(i), \beta(j)), & \\
\quad \text{if } (\text{last}(\text{nextNo}(\alpha(i)))<s1 \text{ and } s2<\text{first}(\text{prevNo}(\beta(j)))) & \qquad (10a)\\
S(\text{nextNo}(\alpha(i)), \text{prevNo}(\beta(j))) + \delta(\alpha(i), \beta(j)), & \\
\quad \text{if } (\text{last}(\text{nextNo}(\alpha(i)))<\text{last}(\text{prevNo}(\beta(j)))<s1\text{or} & \\
\quad s2<\text{first}(\text{nextNo}(\alpha(i)))<\text{fisr}(\text{prevNo}(\beta(j)))) & \qquad (10b)\\
\max\{\max_{i<k<\text{fisrt}(\text{nextStart}(\beta(k)))<s1}[S(\alpha(i),\beta(k))+QK(\text{nextStart}(\beta(k)),\beta(j),s1,s2)], & \qquad (11a)\\
\quad \max_{s2<k<\text{fisrt}(\text{nextStart}(\beta(k)))<j}[QK(\alpha(i),\beta(k),s1,s2)+S(\text{nextStart}((\beta(k)),\beta(j))], & \qquad (11b)\\
\quad \max_{i<k<s1\text{and } s2<\text{first}(\text{nextStart}(\beta(k)))<j}[S(\alpha(i),\beta(k)) + S(\text{nextStart}(\beta(k), \beta(j))] & \qquad (11c)\\
\quad \}, & \\
\max\ \{\max_{\substack{i<i2<j2<s1\\ s2<j3<=j}}[P(\alpha(i), \beta(j_2),\alpha(i_2))+Q(s2+1,\beta(j), \beta(j_3)) + & \\
\qquad S(\text{nextStart}(\beta(j_2)), s1-1)+\delta(\alpha(i_2), \beta(j_3))] & \qquad (12a)\\
\quad \max_{\substack{s2<i2<j2<j\\ i=<i3<s1}}[P(\alpha(i_2), \beta(j), \beta(j_2))+Q(\alpha(i), s1-1, (\alpha(i_3)) + & \\
\qquad S(s2+1, \text{prevNo}(\alpha(i_2))+\delta((\alpha(i_3), \beta(j_2))] & \qquad (12b)\\
\quad \} &
\end{cases}$$

Note: $s1=\text{first}(\alpha(i_1))$ or $\text{first}(\beta(j_1))$, $s2 = \text{last}(\alpha(i_1))$ or $\text{last}(\beta(j_1))$. s1 and s2 are constant.

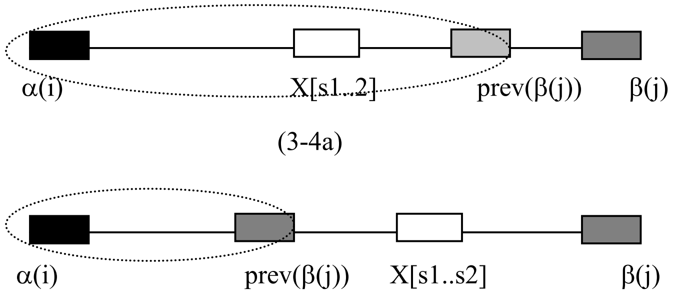
The most important objective of this algorithm is to guarantee that the indices used for a recursive call are smaller than s1 and larger than s2 so that the pseudoknot-base-regions $\alpha(i_1)$ or $\beta(j_1)$ are not overlapped by other base regions throughout the calculation. Formulas (8a) and (8b) calculate the score of the structure for $X[\alpha(i)..\beta(j)]$ excluding base region X[s1..s2] when the non-matching region α is not considered (Fig. 3-3). Formula (8a) handles the case that next(α(i)) returns base regions starting at position < s1; formula (8b) deals with the case that next(α(i)) returns base regions starting at position > s2; Similarly, formula (9a) and (9b) calculate the score of the structure for $X[\alpha(i)..\beta(j)]$ excluding base region X[s1..s2] when the non-matching region β(j) is excluded (Fig. 3-4). Formula (9a) handles the case that prev(β(j)) returns base regions ending at position < s1; formula (9b) handles the case that prev(β(j)) returns base regions ending at position > s2.



(3-3a)

(3-3b)

Fig. 3-2. For subsequence $X[\alpha(i)..\beta(j)]$, formulas (8a and 8b) calculate the score of the structure for $X[\alpha(i)..\beta(j)]$ excluding base region X[s1..s2] when the non-matching region α(i) is not considered. The parts that are used for recursive calls are indicated by the circles, where s1 and s2 are defined in table 3-2.

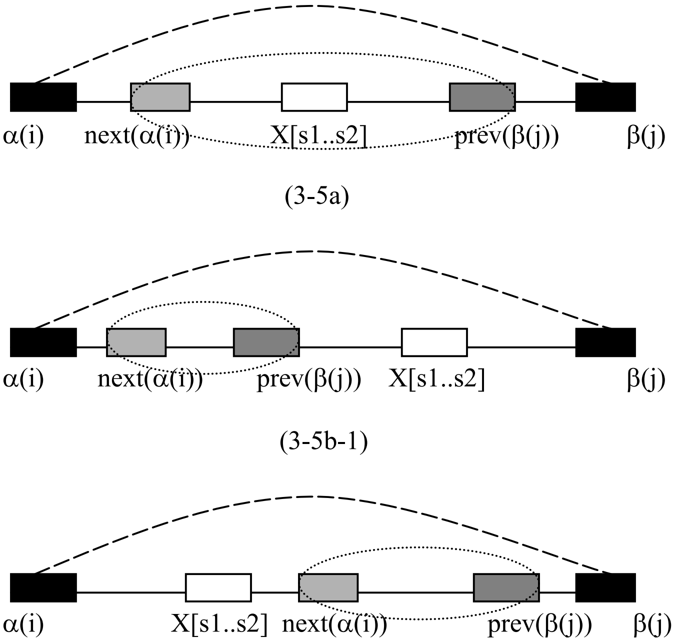


(3-4a)

(3-4b)

Fig. 3-3. For subsequence $X[\alpha(i)..\beta(j)]$, formulas (9a) and (9b) calculate the score of the structure for $X[\alpha(i)..\beta(j)]$ excluding base region X[s1..s2] when the non-matching region β(j) is excluded. The parts that are used for recursive calls are indicated by the circles.

Formula (10a) and (10b) calculate the score of the structure for X[α(i)..β(j)] excluding base region X[s1..s2] when the matchable region β(j) is included (Fig. 3-5). Formula (10a) handles the case that nextNo(α(i)) returns base regions starting at position < s1 and prev(β(j)) returns base regions ending at position > s2; formula (10b) handles the case that nextNo(α(i)) returns base regions starting at position > s2 or prev(β(j)) returns base regions ending at position < s1.



(3-5a)

(3-5b-1)

(3-5b-2)

Fig. 3-4. For subsequence X[α(i)..β(j)], formulas (10a and 10b) calculate the score of the structure for [α(i)..β(j)] excluding base region X[s1..2] when the matchable region α(i) and β(j) are included. The parts that are used for further recursive calls are indicated by the circles.

Formula (11a), (11b) and (11c) combines two optimal substructures for X[α(i).. β(k)] and X[nextStart(β(k))..β(j)]. Formula (11a) handles the case that the partitioning point is located at positions < s1; formula (11b) handles the case that the partitioning point is located at position > s2; formula (11c) deals with the case that the base region X[s1..s2] is located between base region β(k) and nextStart(β(k)) (see Fig. 3-6).

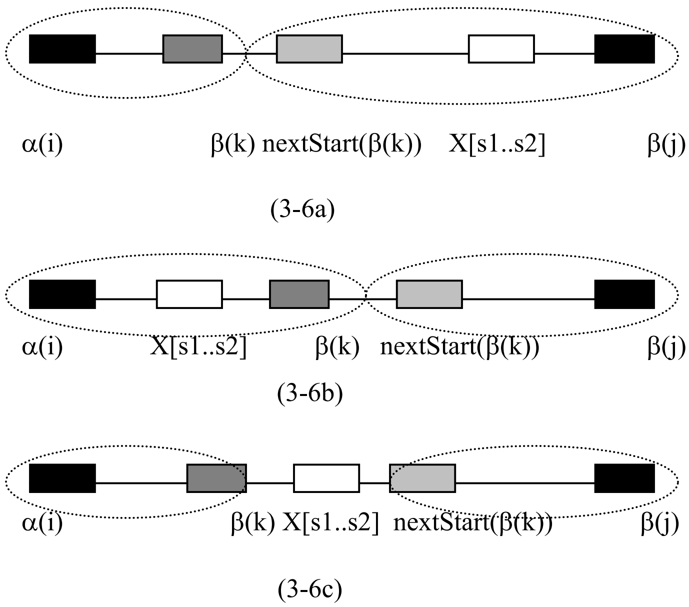


Fig. 3-5. For subsequence X[$\alpha(i)..\beta(j)$], formulas (11a), (11b) and (11c) combines two optimal substructures for X[$\alpha(i)..\beta(k)$] and X[nextStart($\beta(k)$)..$\beta(j)$]. The circles indicate the parts that are used for further recursive calls.

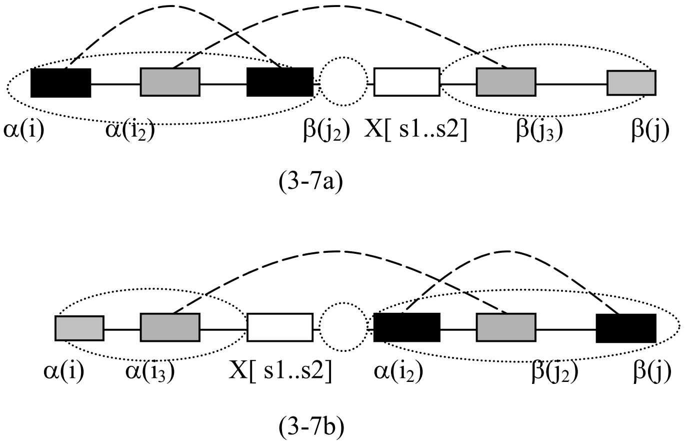


Fig. 3-6. For subsequence X[$\alpha(i)..\beta(j)$], formulas (12a and 12b) looks for (a) a pseudoknot formed by a double helix h($\alpha(i)$, $\beta(j_2)$) and a second double helix h($\alpha(i_2)$, $\beta(j_3)$), (b) a pseudoknot formed by a double helix h($\alpha(i_2)$, $\beta(j)$) and a second double helix h($\alpha(i_3)$, $\beta(j_2)$). The circles indicate the parts that are used for further recursive calls.

In this modified algorithm, we continue to look for a pseudoknot and calculate the maximum score for the possible pseudoknot by formulas (12a) and (12b). The pseudoknots are assumed to occur in two situations here, one is that $\alpha(i)$ forms a double helix with a base region $\beta(j_2)$, where last($\beta(j_2)$) < s1 and

$\alpha(i_2)$ forms a second double helix with $\beta(j_3)$, where last(α(i))<first($\alpha(i_2)$)<last($\alpha(i_2)$)<first($\beta(j_2)$), s2<first($\beta(j_3)$)< last($\beta(j_3)$)<=last(β(j)) (Formula (12a), Fig. 3-7a). The other is that β(j) form a double helix with a base region $\alpha(i_2)$, where s2<first($\alpha(i_2)$) and $\alpha(i_3)$ forms a second double helix with $\beta(j_2)$, where first(α(i)) =< first($\alpha(i_3)$)<last($\alpha(i_3)$)<s1, last($\alpha(i_2)$)<first($\beta(j_2)$)<last($\beta(j_2)$)<first(β(j)) (Formula (12b), Fig. 3-7b). The subsequence X[α(i)..β(j)] is partitioned into three parts which are indicated by three circles, for the recursive calculation of the overall log odds score. Because the algorithm looks for pseuoduknots in these two situations, more complex pseudoknot structures could be missed in the calculation. As in the calculation of formulas (3) and (5), whenever matchable base regions α(i) and β(j) are considered in the calculation of the formula (10a-b) and (12a-b), the complete helix $\alpha\beta$ and its three subhelices are tried. The one that results in the structure with highest log odds score is used. The algorithm runs in time $O(M^3)$, where M is the number of the base pairing regions.

CHAPTER 4

IMPLEMENTATION AND PRELIMINARY TESTS

We have implemented the algorithm in C++ on the UNIX/Solaris platform of a SUN workstation. The input to the program is a standard FASTA formatted RNA sequence, a 1x5 base probability matrix, a 5x5 base-pairing matrix and several RNA structure parameters (4 bases plus gaps). The probability data is used to calculate the log odds score for each double helix.

In the implementation of mBLAST-PK algorithm, two techniques were used to reduce the space and time usage. One technique is that we use arrays of lists instead of high dimensional matrices to store the information of base regions and the scores for the pseudoknots. The list of all of base regions is stored in a one-dimensional array. Each element in the array represents a cell of $\alpha(i)$ and references a list of nodes, with each node containing the information of a base region $\beta(j)$ that can form a double helix with base region $\alpha(i)$. This implementation not only saves space, but also reduces the running time, because, when we calculate formulas (5), (12a) or (12b), it is unnecessary to check all the base regions for region $\alpha(i)$, $\alpha(i_1)$ and $\beta(j)$ to find the pairing region $\beta(j_1)$, $\beta(j_2)$ and $\alpha(i_1)$. We only need to check the base regions stored in the lists referenced by the cell of $\alpha(i)$, the cell of $\alpha(i_1)$ and the cell of $\beta(j)$. The value of $QK(\alpha(i), \beta(j), s1, s2)$ is stored in a two dimensional array of lists in which the cell of $\alpha(i)$ and $\beta(j)$ references to a list of nodes. Each node contains the score, trace back pathway and pseudoknot information with specified s1 and s2. This greatly simplifies the storage of data.

To avoid repeatedly calculating the score for the same subsequence, the modified Nussinov algorithm is implemented with a memoized recursive algorithm, a variation of dynamic programming. A top-down memoization recursive algorithm memoizes the solution of a subsequence the first time it is completed. The solution can be simply looked up and returned when it is needed at a subsequent time. The memoization recursive algorithm offers advantages over the bottom-up dynamic programming since only the scores for indices of $\alpha(i)$ and $\beta(j)$ that are valid, need to be calculated.

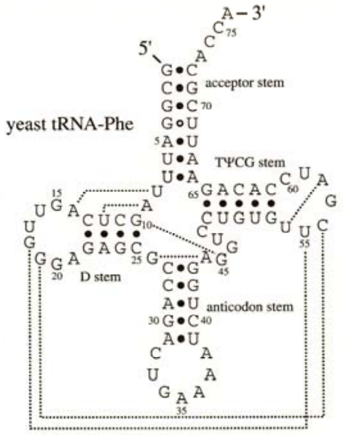
To evaluate the success of this algorithm, we ran our program on a set of tRNA sequences and a set of tm RNA sequences. Our predicted structures for these sequences have been classified into three categories. The first category includes the sequences for which our predicted structures are exactly the

Table 4-1 Running time comparisons between mBlast-PK algorithm and other pseudoknot structure prediction algorithms-PKNOTS and PCGS (parallel communicating grammar systems). The numbers in parenthesis are the number of double helices regions identified by the mBLAST algorithm.

Sequence length	Algorithm	Running time	References
84 nucleotides	PKNOTS	47 min	Rivas and Eddy 1999 [15]
105	PKNOTS	235 min	Rivas and Eddy 1999 [15]
100	PCGS	60 min	Cai et al. [5]
150 (70)	mBLAST-PK	0.05 min	
204 (144)	mBLAST-PK	0.9 min	

same as their known structure. The second category includes the sequences for which our predicted structures include the same set of double helices, but with errors in the boundaries of some base regions or with an extra double helix. The third category includes the sequences for which our predicted structures missed one or more double helices. The accuracy of the prediction is calculated as the ratio of the number of sequences in the first category and the secondary categories to the total number of the sequences tested.

We first ran our program on a yeast tRNA-phe sequence (Figure 4) and all the available E. coli-K12 tRNA sequences (88 tRNA sequences from the tRNA database (http://rna.wustl.edu/GtRDB/Eco/)). The



```
predicted structure for tRNA-phe length 76
GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAAAUCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA
AAAAAAA--BBBB--------bbbb-CCCCC-------ccccc-----ZZZZZ-------zzzzzaaaaaaa----
```

Fig. 4. The stem-loop structure of yeast tRNA-phe [6] and the predicted structure for it.

Table 4-2. Some typical outcomes of the prediction test on 88 E. coli-k12 tRNA sequences. Under each sequence is the correct stem-loop structure followed by the predicted structures. The correct structures (in the first category) as in sequence trna70 totaled 83% of the prediction. The incorrect predictions in the third category are shown as in sequence trna52.

```
Ecoli-K12.trna70 length 76
GGGGCUAUAGCUCAGCUGGGAGAGCGCUUGCAUGGCAUGCAAGAGGUCAGCGGUUCGAUCCCGCUUAGCUCCACCA
AAAAAAA--BBBB--------BBBB-CCCCCCC---CCCCCCC-----ZZZZZ-------ZZZZZAAAAAAA----
AAAAAAA--BBBB--------BBBB-CCCCCCC---CCCCCCC-----ZZZZZ-------ZZZZZAAAAAAA----

Ecoli-K12.trna52 length 87
GCGAAGGUGGCGGAAUUGGUAGACGCGCUAGCUUCAGGUGUUAGUGUCCUUACGGACGUGGGGGUUCAAGUCCCCCCCCCUGCACCA
AAAA-AA--BBB-----------BBB-CCCCC------CCCCC--ZZZZ----ZZZZ--DDDDD-------DDDDDAA-AAAA----
--AAAA-BBBBB-------------BBBBB-AAAA-CCCCCCCCCZZZZZ---ZZZZZ-DDDDD-------DDDDD--CCCCCCCC-
```

secondary structure of a tRNA sequence only consists of stem-loops. Compared with Cove-predicted tRNA secondary structures rendered by NAVIEW (c) (http://rna.wustl.edu/GtRDB/Eco/Eco-structs.html), our predicted structures for the tRNA sequences have been classified into two categories. The first category includes 73 tRNA sequences for which our predicted structures are exactly the same as the Cove-predicted. The second category includes 15 the tRNA sequences for which our predicted structures are not the same as the Cove-predicted. The accuracy of our prediction for these tRNA sequences is 73/88 = 83% (Table 4-1).

We further tested our program on a set of 80 tmRNA sequences from the tmRNA database (http://psyche.uthct.edu/dbs/tmRNA). In the first trial, if any double helix in the structure is assumed to have at least 3 base pairs, mismatch extension of hits was not performed and subhelices were not considered in the score calculation. Our program predicted pseudoknots in 76 sequences of the 80 sequences. For 41 of the 76, the program correctly predicted the pseudoknot with the structure and exact base-pairing regions (the first category). For 13 sequences, the overall structure was correctly predicted with errors in the length of the base regions (the second category). The algorithm predicted the pseudoknots incorrectly for the remaining 28 sequences (the third category). The program could not predict the pseudoknots successfully for 22 of the 28 because these sequences have bulges or noncanonical base pairs in their structures. The pseudoknots of the remaining 6 sequences were predicted

Table 4-3. The improvement of the prediction accuracy for psedoknot structures using different input parameters. The prediction accuracy reaches 81% when the minimal length of a double helix is either 3 or 4, match and mismatch extensions are allowed in the mBLAST algorithm and subhelices are considered in the extended Nussinov algorithm.

Prediction accuracy	the min double helix length	match extension	mismatch extension	sub-helices considered
(41+13)/80 = 68%	3	yes	no	no
(45+16)/80 = 75%	4	yes	no	no
(48+19)/80 = 81%	3 or 4	yes	yes	yes

incorrectly because these sequences have a pseudoknot with only 1 or 2 base pairs. This is the same reason that the program also missed predicting pseudoknots in the remaining 4 of 80 tmRNA sequences. The correct and nearly correct structures account for (41+13)/80 = 68%.

Since most of the double helices in the tested tmRNA have at least 4 base pairs, we tested our program on the tmRNA data assuming the minimum length of a stem is 4, without performing mismatch extension of hits and without considering subhelices in the score calculation. In that case, the percentage of the correct structure (the first category) and nearly correct structures (the second category) increased up to (45+16)/80 = 75%. Finally, we tested our program on the same tmRNA sequences performing extension in the mBLAST algorithm and considering subhelices in the score calculation, while the minimal length was assumed to be either 3 or 4. Using the optimized structure parameters, the program predicted the structure correctly for 48 sequences and nearly correctly for 19 sequences. The percentage of the correct and nearly correct structures increased up to (48+19)/80 = 81%. The algorithm failed to predict the pseudoknots correctly for the remaining 13 sequences, of which 6 sequences have a pseudoknot with only 1 or 2 base pairs.

In the examples shown in Table 4-2, sequence 37 demonstrate an exactly correct prediction; sequence 8 shows a prediction of a pseudoknot, but with an extra double helix; sequence 35 shows a prediction that is structurally correct, but with errors in the regions; sequence 7 demonstrates failure to predict the structure.

Table 4-4. Some typical outcomes of the prediction test on tmRNA sequences. Under each sequence is the correct pseudoknotted structure followed by the predicted structures. The correct structures as in sequence 37 (the first category) plus nearly correct predictions (the second category), similar to sequence 8, totaled 81% of the prediction.

Sequence pk2-37
```
CGUUGCAGCAGUCGGUCAAUGGGCUGUGUGGCGAAAGCCACCGCAACGUCAUCUUACAUUGA
AAAAAA--HHHHH--BBBBBBHHHHH-LLLLL----LLLLL-AAAAAA--------BBBBBB
AAAAAA--HHHHH--BBBBBBHHHHH-LLLLL----LLLLL-AAAAAA--------BBBBBB
```

Sequence pk2-8
```
CCCUCUGCCCGGAUUUGUCUGUGGAUCCGGAGCCGAAAGGCGCGCGGAGGGUCAUGAAACACGGA
AAAAAAAAHHHHHHH--BBBBBBHHHHHHH-------------AAAAAAAA--------BBBBBB
AAAAAAAAHHHHHHH--BBBBBBHHHHHHH-QQQ----QQQ--AAAAAAAA--------BBBBBB
```

Sequence pk2-35
```
CGCUGCACUGAUCUGUCCUUGGGUCAGGCGGGGAAGGCAACUUCACAGGGG
CCCCCC-BBBBB---AAAAABBBBBBCCCCCC-------------AAAAA-
--CCCC-BBBB-----AAAAA--BBBBCCCC--DDDD---DDDD-AAAAA-
```

Sequence pk1-7
```
GUAUGAUUCCACCGGIGGUUUUUGCCAUAUGGAUCA
AAAAAAA--------BBBB----AAAAAAA--BBBB
---AAAAAA------BBBB----BBBB---AAAAAA
```

CHAPTER 5

DISCUSSIONS AND CONCLUSIONS

RNA secondary structure including pseudoknots is computationally difficult to predict. Almost all existing algorithms for optimal pseudoknot prediction entail O(n^6) running time and O(n^4) memory space even for restricted categories of pseudoknots, making pseudoknot prediction unrealistic for RNA of more than 100 nucleotides.

We introduced a novel algorithm mBLAST-PK for RNA pseudoknot prediction that can substantially reduce the computational costs. Following the idea of the BLAST algorithm, the new algorithm preprocesses the RNA sequence to find all base pairing regions in the sequence based on a modified BLAST process. It then non-trivially extends Nussinov folding to calculate the optimal structure including pseudoknots. Our experiments with the implemented algorithm showed its effectiveness and predicted the RNA stem-loop structures in bacterial tRNA sequences at about 83% accuracy, and the RNA pseudoknot structures in bacterial tmRNA sequences at about 81% accuracy. The running time and memory space consumption by the algorithm are both reduced by at least two orders of magnitude, making the task of pseudoknot prediction routine on desktop computers. Currently, the mBLAST -PK algorithm is applied to single RNA structure prediction; however, potentially the algorithm may provide feasible solutions to multiple-sequence consensus structure prediction.

To speed up the prediction process, a double helix is assumed to have at least 3 base pairs and the non-canonical base pair G-U is only allowed under certain conditions. These assumptions and the way in which the pseudoknot is computed imply that the algorithm is not able to predict all kinds of RNA structures that may occur in reality. The algorithm can predict a bulge structure under the condition that the base-pairing regions on both sides of a bulge are longer than 2; the algorithm is not able to predict the bulge structure without this condition.

The accuracy might be enhanced by allowing more flexible conditions for when a hit can extend or permitting the G-U base pair to occur in the mBLAST algorithm. More flexible conditions may slow down the prediction process because many more base regions will be identified in the preprocessing. Our experiments also showed that the prediction accuracy of our program decreases as the length of the testing RNA sequence increases. In our current implementation of this algorithm, the log odds score is calculated based on the base probabilities and base pairing probabilities, which were obtained from the training tmRNA sequence dataset. No further biological information is involved in the prediction process. Additional information could be incorporated into the rules to yield a biologically meaningful folding of a molecule [23], especially for long RNA sequences.

REFERENCES

1. Akutsu, T. (2000) Dynamic programming algorithms for RNA secondary prediction with pseudoknots. Discrete Applied Mathematics, 104, pp. 45-62.

2. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool. J. Mol. Biol., 215, 403-10.

3. Brown, M. and Wilson, C. (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. Pacific Symposium on Biocomputing 1996.

4. Cary, R.B. and Stormo, G.D. (1995) Graph-theoretic approach to RNA modeling using comparative data. ISMB, 95, 75-80.

5. Cai, L., Malmberg, R. L. and Wu. Y. (2003) Stochastic modeling of RNA pseidoknotted structures: a grammatical approaches. Bioinformatics, 19, 166-173.

6. Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G. 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. Nucl. Acids Res., 22, 2079-2088.

8. Felden, B., Massire,C., Westhof,E., Atkins., J.F. and Gesteland, R.F. (2001) Phylogenetic analysis of tmRNA genes within a bacterial subgroups reveals a specific structural signature. Nucleic Acids Res., 29, 1602-1607.

9. Gorodkin, J., Heyer, L.J., and Stormo, G.D. (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences, Nucleic Acids Research, 25(18), pp. 3724-3732.

10. Yuh-Jyh Hu. (2003) GPRM: a genetic programming approach to finding common RNA secondary structure elements. Nucleic Acids Res. 31 (13), 3446–3449

11. Ji, Y. and Stormo, G.D. (2003) A new approach to identify conserved RNA secondary structural motifs in homologous sequences. Poster, The fourth Georgia Tech and UGA international Conference on Bioinformatics, Nov. 13-16, Atlanta, Georgia, USA.

12. Lyngsø, R.B. and Pedersen, C.N.S. (2000) RNA Pseudoknot Prediction in EnergyBased Models. Journal of Computational Biology, 7(3/4), pp. 409-428.

13. Nussinov, R. and Jacobson, A.B. (1980) Fast algorithm for predicting the secondary structure of single-stranded *RNA*. Proceedings of the National Academy of Sciences of the USA, 77, pp. 6309-6313.

14. Paillart,J.C., Skripkin,E., Ehresmann,B., Ehresmann,C. and Marquet,R. (2002) *In vitro* evidence for a long range pseudoknot in the 5'-untranslated and matrix coding regions of HIV-1 genomic RNA. J. Biol. Chem., 277, 5995–6004.

15. Rivas, E. and Eddy, S. R. (1999) Dynamic programming algorithm for RNA structure prediction including pseudoknots. Journal of Molecular Biology, 285(5), pp. 2053-2068.

16. Rivas, E and Eddy, S.R. (2000) The language of RNA: A formal grammar that includes pseudoknots. Bioinformatics 16, 334-340.

17. Sankoff, D. (1985) Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems, SIAM Journal on Applied Mathematics, 45(5), pp. 810-825.

18. Shapiro, B.A. and Wu, J-C. (1996) An annealing mutation operator in the genetic algorithms for RNA folding. CABIOS, 12(3), 171-180.

19. Tabaska, J. E., Cary, R. B., Gabow, H. N. and Stormo, G. D. (1998) An RNA folding method capable of identifying pseudoknots and base triples. Bioinformatics. 14:691-699

20. Tanaka, Y., Hori, T., Tagaya, M., Sakamoto, T., Kurihara, Y., Katahira, M. and Uesugi, S. (2002) Imino proton NMR analysis of HDV ribozymes: nested double pseudoknot structure and Mg^{2+} ion-binding site close to the catalytic core in solution. Nucleic Acids Res., 30(3), 766-74.

21. Tinoco, I. Jr. and Bustamante, C. (1999) How RNA Folds. J. Mol. Biol., 293, 271-281.

22. Uemura, Y., Hasegawa, A., Kobayashi, Y., and Yokomori, T. (1999) Tree adjoining grammars for RNA structure prediction. Theoretical Computer Science, 210, pp. 277-303.

23. Zuker, M. and Stiegler, P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res., 9, pp. 133-148.

24. Zuker, M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res., 31 (13), 3406-15.

25. Bol, J.F. and Olsthoorn, R.C.L. Sequence comparison and secondary structure analysis of the 3' noncoding region of flavivirus genomes reveals multiple pseudoknots. RNA, 2001, 7:1370.

26. Gautheret, D. and Gutell, R.R. Inferring the conformation of RNA base pairs and triples from patterns of sequence variation. Nucleic Acids Res., 1997, 25:1559.

27. Chen, J. Le, S. and Maizel, J.V. A procedure for RNA pseudoknot prediction. 1992, 8: 243.

28. Gultyaey, A.P. The computer simulation of RNA folding involving pseudoknot formation. Nucleic Acids Res., 1991,19:2489

29. Dirks, R.M. and Pierce, N.A. A Partition Function Algorithm for Nucleic Acid Secondary Structure Including Pseudoknots. J. Comput. Chem., 2003, 24:1664.

30. Prederique Galisson. The fasta and blast programs. http://bioweb.pasteur.fr/seqanal/blast/ blast_fasta-uk.ps. 2000. p7.