Journal of Combinatorial Optimization 3, 321–336 (1999)

Approximation and Exact Algorithms for RNA Secondary Structure Prediction and Recognition of Stochastic Context-free Languages*

TATSUYA AKUTSU takutsu@ims.u-tokyo.ac.jp
Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Abstract. For a basic version (i.e., maximizing the number of base-pairs) of the RNA secondary structure prediction problem and the construction of a parse tree for a stochastic context-free language, $O(n^3)$ time algorithms were known. For both problems, this paper shows slightly improved $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$ time exact algorithms, which are obtained by combining Valiant's algorithm for context-free recognition with fast funny matrix multiplication. Moreover, this paper shows an $O(n^{2.776} + (1/\epsilon)^{O(1)})$ time approximation algorithm for the former problem and an $O(n^{2.976}\log n + (1/\epsilon)^{O(1)})$ time approximation algorithm for the latter problem, each of which has a guaranteed approximation ratio $1 - \epsilon$ for any positive constant ϵ, where the absolute value of the logarithm of the probability is considered as an objective value in the latter problem. The former algorithm is obtained from a non-trivial modification of the well-known $O(n^3)$ time dynamic programming algorithm, and the latter algorithm is obtained by combining Valiant's algorithm with approximate funny matrix multiplication. Several related results are shown too.

Keywords: computational biology, RNA secondary structure prediction, stochastic context-free grammar, approximation algorithms

1. Introduction

RNA secondary structure prediction is an important problem in *computational biology* and thus many computational studies have been done. This is a problem of, given an RNA sequence of length n, finding its correct secondary structure (an out-planar graph like structure, see figure 1). Usually, RNA secondary structure prediction is modeled as a *free-energy minimization* problem (Setubal and Meidanis, 1997; Waterman, 1995). For this problem, Waterman and Smith (1978), and Zuker and Stiegler (1981) proposed simple DP (*dynamic programming*) algorithms. The time complexities of those DP algorithms were $O(n^3)$ if we ignore the *destabilizing energy due to loop regions*, otherwise they were at least $O(n^4)$.

In a basic and simplest version, free-energy minimization of an RNA secondary structure is defined as a problem of *maximizing the number of complementary base pairs* (see figure 1),

*A preliminary version of this paper appeared in ISAAC'98 (Ninth International Symposium on Algorithms and Computation).

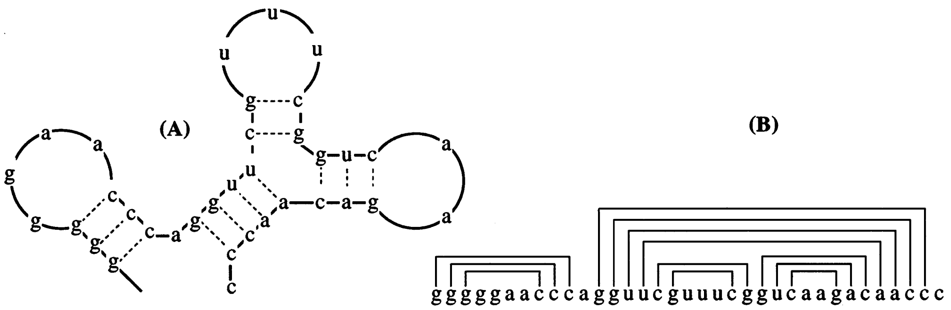

Figure 1. Two representations of RNA secondary structure: (A) ‘Clover leaf’ representation similar to real structure; (B) Sequence is represented on the horizontal axis. In a basic version, RNA secondary structure prediction is defined as a problem of maximizing the number of base pairs (i.e., (a, u) and (g, c) pairs) which do not intersect each other.

which is denoted by $\mathbf{RNA}_0$ in this paper. Even for $\mathbf{RNA}_0$, only an $O(n^3)$ time simple DP algorithm had been known (Setubal and Meidanis, 1997; Waterman, 1995).

Although no further improvement had been done on *global free-energy minimization*, several important improvements have been done for finding *locally stabilizing substructures* in an RNA secondary structure (Galil and Park, 1992; Waterman, 1995). Waterman and Smith (1986) developed an $O(n^3)$ time algorithm for an arbitrary destabilizing energy function. Kanehisa and Goad (1982) developed an $O(n^2)$ time algorithm for a *linear* destabilizing energy function. Eppstein et al. (1988) developed an $O(n^2 \log^2 n)$ time algorithm for a concave or convex destabilizing energy function. Slight improvements (Eppstein et al., 1994; Larmore and Schieber, 1991) have been done for the same case.

On the other hand, *stochastic context-free grammars* (SCFG, in short) have been applied to RNA secondary structure prediction (Sakakibara et al., 1994), in which a parse tree with the highest probability (an *optimal parse tree*) corresponds to an RNA secondary structure with the minimum free-energy (an *optimal RNA secondary structure*). However, to the best of our knowledge, only an $O(n^3)$ time algorithm was known for the construction of an optimal parse tree for SCFG.

In this paper, we first show a slightly improved $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$ time exact algorithm for the construction of an optimal parse tree for SCFG, which can also be applied to $\mathbf{RNA}_0$. This algorithm is a simple combination of Valiant’s algorithm for *context-free recognition* (Valiant, 1975) with a fast algorithm for *funny matrix multiplication* (Fredman, 1976; Takaoka, 1992). Note that funny matrix multiplication is, given $p \times q$ real matrix $X = (x_{ij})$ and $q \times r$ real matrix $Y = (y_{ij})$, to compute $p \times r$ matrix $Z = (z_{ij})$ such that $z_{ij} = \max_{1 \le k \le q}(x_{ik} + y_{kj})$. In this paper, we denote funny matrix product of X and Y by $X \odot Y$. For several problems such as the all-pairs shortest path problem (Alon et al., 1991; Takaoka, 1992) and the maximum subarray problem (Tamaki and Tokuyama, 1998), the fastest algorithms were obtained using fast funny matrix multiplication.

Next, we show the main result of this paper: an $O(n^{2.776} + (1/\epsilon)^{O(1)})$ time approximation algorithm for $\mathbf{RNA}_0$ which always outputs an RNA secondary structure with the score at least $1 - \epsilon$ of the maximum, where ϵ is any positive constant number and the score denotes the number of complementary base pairs in $\mathbf{RNA}_0$. Although this algorithm can be considered as a PTAS (polynomial time approximation scheme), it is different from usual PTAS since the problem is not NP-hard but belongs to P. This algorithm is a combination of an approximation algorithm $\mathcal{A}_{\text{approx}}$ and an exact algorithm $\mathcal{A}_{\text{exact}}$, where $\mathcal{A}_{\text{approx}}$ is obtained by modifying the original $O(n^3)$ time DP algorithm for $\mathbf{RNA}_0$, and $\mathcal{A}_{\text{exact}}$ is obtained by combining Valiant's algorithm with fast funny matrix multiplication. Although Tamaki and Tokuyama (1998) developed an $o(n^3)$ time approximation algorithm for the maximum subarray problem based on fast funny matrix multiplication, their technique could not be applied to $\mathbf{RNA}_0$ and thus a new technique was introduced. Moreover, although $\mathcal{A}_{\text{approx}}$ is obtained by modifying the original DP algorithm, the modification and the analysis are non-trivial. Then, we extend the technique used in $\mathcal{A}_{\text{approx}}$ for more practical versions of RNA secondary structure prediction.

We also modify the technique for the construction of an optimal parse tree for SCFG, and obtain an $O(n^{2.976} \log n + (1/\epsilon)^{O(1)})$ time approximation algorithm, which always outputs a parse tree with the score at least $1 - \epsilon$ of the optimal, where ϵ is any positive constant number and we consider the absolute value of the logarithm of the probability as the score.

2. RNA secondary structure and SCFG

In this section, we formally define the problem $\mathbf{RNA}_0$ and then describe a relationship between $\mathbf{RNA}_0$ and SCFG.

2.1. A basic version of RNA secondary structure prediction

Let $A = a_1, a_2, \ldots, a_n$ be an *RNA sequence*. That is, A is a string over an alphabet $\Sigma = \{\mathtt{a}, \mathtt{u}, \mathtt{g}, \mathtt{c}\}$. A pair of residues (letters) (x, y) is called a (*complementary*) *base pair* if $\{x, y\} = \{\mathtt{a}, \mathtt{u}\}$ or $\{x, y\} = \{\mathtt{g}, \mathtt{c}\}$. Although $\{\mathtt{g}, \mathtt{u}\}$ is sometimes treated as a base pair (Waterman, 1995), all the results in this paper are valid even if $\{\mathtt{g}, \mathtt{u}\}$ is treated as a base pair, except that approximation ratio $1/2$ in Theorem 4 is replaced by $1/3$. A set of pairs of indices $M = \{(i, j) \mid 1 \le i < j \le n, (a_i, a_j) \text{ is a base pair}\}$ is called an *RNA secondary structure* if no distinct pairs $(a_i, a_j), (a_h, a_k)$ in M satisfy $i \le h \le j \le k$ (see figure 1). The score of M is defined as the number of base pairs in M (i.e., $|M|$), and denoted by *score*(M). Then, $\mathbf{RNA}_0$ is defined as follows: given an RNA sequence $A = a_1, a_2, \ldots, a_n$, to find an RNA secondary structure M with the maximum score. In $\mathbf{RNA}_0$, such a structure is also called an *optimal RNA secondary structure*, and denoted by $OPT_0(A)$.

It is well known that the score of $OPT_0(A)$ can be computed in $O(n^3)$ time using the following simple DP procedure (denoted by $\mathcal{DP}_0$):

$$S(i, j) = \max \begin{cases} S(i+1, j-1) + \mu(a_i, a_j), \\ \max\limits_{i<k\le j} \{S(i, k-1) + S(k, j)\}, \end{cases}$$

where we let $S(i, j) = 0$ for all $i \ge j$, and $\mu(x, y) = 1$ if (x, y) is a base pair, otherwise

$\mu(x, y) = 0$. Note that the score of $OPT_0(A)$ is given by $S(1, n)$. $OPT_0(A)$ can also be obtained in $O(n^3)$ time using the *traceback* technique (Waterman, 1995). Similarly, we only describe the procedures for computing scores or free-energies in this paper, all of which can be modified for computing secondary structures or parse trees without increasing the orders of the time complexities using the traceback technique.

2.2. *SCFG and its relationship with* $\mathbf{RNA}_0$

A *stochastic context free-grammar* (SCFG) is a context-free grammar in which every production rule has an associated probability value. We denote the associated probability for a production $X \rightarrow \alpha$ by $P(X \rightarrow \alpha)$. Usually, $\sum P(X \rightarrow \alpha) = 1$ should hold for each non-terminal symbol X, where the sum is taken over all rules whose left hand sides are X.

The *probability of a parse tree* is defined by the product of the probabilities of the productions used to generate the sequence. The *probability of a sequence s* is the sum of probabilities over all possible parse trees that could generate s. An *optimal parse tree* for a sequence s is a parse tree for s with the highest probability.

Several papers (Sakakibara et al., 1994; Uemura et al., 1995) pointed out a relationship between RNA secondary structure and SCFG. Based on them, we can associate the context-free grammar shown in Table 1 with $\mathbf{RNA}_0$, where a score is associated for each production rule instead of a probability, X, Y, Z denote the same non-terminal symbol, and *score*(X) denotes the score of a node in a parse tree (or, a subtree in a parse tree) corresponding to the production rule. In this case, a parse tree whose root has the maximum score (among all parse trees) corresponds to an optimal secondary structure in $\mathbf{RNA}_0$. Note that, in this case, the score of a parse tree is not the product of the probabilities, but the sum of the scores of productions used to generate the sequence. However, a parse tree with the highest probability is equal to a parse tree with the maximum score if we assign score log p (≤ 0) to each production rule having probability p.

Table 1. Stochastic context-free grammar associated with $\mathbf{RNA}_0$, where X, Y, Z denote the same non-terminal symbol.

	Score of rule	Score (X)
$X \rightarrow \epsilon$	0	0
$X \rightarrow \mathtt{a}$	0	0
$X \rightarrow \mathtt{u}$	0	0
$X \rightarrow \mathtt{g}$	0	0
$X \rightarrow \mathtt{c}$	0	0
$X \rightarrow YZ$	0	$score(Y) + score(Z)$
$X \rightarrow \mathtt{a}Y\mathtt{u}$	1	$score(Y) + 1$
$X \rightarrow \mathtt{u}Y\mathtt{a}$	1	$score(Y) + 1$
$X \rightarrow \mathtt{g}Y\mathtt{c}$	1	$score(Y) + 1$
$X \rightarrow \mathtt{c}Y\mathtt{g}$	1	$score(Y) + 1$

3. Exact algorithms

Valiant (1975) developed an $O(n^\omega)$ time algorithm for context-free recognition using a fast boolean matrix multiplication algorithm, where $O(N^\omega)$ denotes the time complexity of the current best algorithms ($\omega = 2.376$ due to Coppersmith and Winograd, 1990) for both the boolean matrix multiplication and the usual (real) matrix multiplication for $N \times N$ matrices. Note that, following the standard convention, we consider a fixed grammar and the size of the grammar is assumed to be a constant. Moreover, we assume without loss of generality that a grammar is given in the *Chomsky normal form*. Valiant's algorithm computes an $n \times n$ matrix $X = (x_{ij})$ for a sequence $s = s_1, s_2, \ldots, s_n$ such that $x_{ij} = 1$ if there is a parse tree for $s_i, s_{i+1}, \ldots, s_{j-1}$, otherwise $x_{ij} = 0$. Valiant proved the following theorem, which also plays an important role in this paper.

Theorem 1 (Valiant, 1975). *Let $M(n)$ be the time complexity of a matrix-multiplication algorithm for $N \times N$ boolean matrices such that there exist constants $K_1 > 2$ and $K_2 > 0$ satisfying*

$$2^{K_1} \cdot M(2^m) \le M(2^{m+1}), \quad (0 \le p < 2^m)(M(2^{m+1}) \le K_2 \cdot M(2^m + p)).$$

Then matrix X can be computed in $O(M(n))$ time.

Modifying Valiant's algorithm, we can obtain an $O(n^\omega)$ time algorithm for computing the probability of a sequence in SCFG.

Theorem 2. *For SCFG the probability of a given sequence s can be computed in $O(n^\omega)$ time.*

Proof: We modify Valiant's algorithm so that x_{ij} represents the probability of a subsequence $s_i, s_{i+1}, \ldots, s_{j-1}$ in SCFG. For that purpose, we replace the boolean matrix multiplication in Valiant's algorithm with the real matrix multiplication. Although other miscellaneous modifications are also required, they are straight-forward. Since the real matrix multiplication for $N \times N$ matrices can be done in $O(N^\omega)$ time and Theorem 1 holds in this case too, the time complexity remains $O(n^\omega)$. □

Modifying Valiant's algorithm, we can also obtain an $O(n^3)$ time algorithm for computing an optimal parse tree in SCFG. However, the time complexity becomes much higher than $O(n^\omega)$. In this case, instead of boolean matrix multiplication or real matrix multiplication, funny matrix multiplication is required because the value which we want to obtain is not the sum of the probabilities but the maximum of the probabilities.

Theorem 3. *For SCFG a parse tree with the highest probability can be computed in $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$ time, where we assume that the logarithm of the probability associated with each production rule can be expressed with $O(\log n)$ bits.*

Proof: In this case, we assign $\log p$ (≤ 0) as a score to each production rule with probability p, and we modify Valiant's algorithm so that x_{ij} denotes the score of an optimal parse tree for subsequence $s_i, s_{i+1}, \ldots, s_{j-1}$ (if there is no parse tree, the score is set to $-\infty$). Then, such matrix X can be computed by replacing the boolean matrix multiplication with the funny matrix multiplication (along with miscellaneous modifications). Since funny matrix multiplication for $N \times N$ matrices takes $O(N^3(\log\log N)^{1/2}/(\log N)^{1/2})$ time (even if negative entries are included) (Takaoka, 1992) and Theorem 1 holds in this case too, the time complexity is $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$. Recall that, once a matrix X is obtained, an optimal parse tree can be obtained using the traceback technique. □

Since an optimal secondary structure in **RNA**$_0$ corresponds to an optimal parse tree in SCFG and the score for each rule is represented with $O(1)$ ($< O(\log n)$) bits, we have:

Corollary 1. **RNA**$_0$ (*i.e., finding an RNA secondary structure with the maximum number of base pairs*) *can be solved in* $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$ *time.*

4. Approximation algorithms for RNA$_0$

4.1. Approximation algorithm with a constant approximation ratio

Here, we give a simple algorithm for **RNA**$_0$ which always outputs an RNA secondary structure with the score at least $1/2$ of the maximum. Although this algorithm is not essential for obtaining an $1-\epsilon$ approximation algorithm for **RNA**$_0$, it is useful to reduce the time complexity of the $1-\epsilon$ approximation algorithm with a constant factor.

Proposition 1. *Suppose that an RNA sequence A consists of letters of* `a` *and* `u`, *and let* #`a` *and* #`u` *be the numbers of occurrences of* `a` *and* `u` *in A respectively. Then, the score of* $OPT_0(A)$ *is equal to* $\min\{\#\mathtt{a}, \#\mathtt{u}\}$. *Moreover,* $OPT_0(A)$ *can be computed in linear time.*

Proof: Using the following procedure, we can compute $OPT_0(A)$ in linear time, which consists of $\min\{\#\mathtt{a}, \#\mathtt{u}\}$ base pairs.

```
Let $S$ be an empty stack;
for $i = 1$ to $n$ do
    if $S$ is empty or $(a_i, top(S))$ is not a base pair then $push(a_i, S)$
    else begin Output $(a_i, top(S))$ as a base pair; $pop(S)$ end
```

□

For an RNA sequence $A = a_1, \ldots, a_n$, let $A(\mathtt{a}, \mathtt{u})$ (resp. $A(\mathtt{c}, \mathtt{g})$) be the subsequence of A consisting of letters of `a` and `u` (resp. `c` and `g`). Then, $score(OPT_0(A))$ is at most the sum of $score(OPT_0(A(\mathtt{a}, \mathtt{u})))$ and $score(OPT_0(A(\mathtt{c}, \mathtt{g})))$. Choosing the better one, we have:

Theorem 4. *For* **RNA**$_0$, *an RNA secondary structure with the score at least* $1/2$ *of the maximum can be computed in linear time.*

4.2. $1 - \epsilon$ approximation algorithm

The $1 - \epsilon$ approximation algorithm is a combination of an exact algorithm $\mathcal{A}_{\text{exact}}$ and an approximation algorithm $\mathcal{A}_{\text{approx}}$: $\mathcal{A}_{\text{exact}}$ is used when *score*$(OPT_0(A))$ is small (precisely, *score* $(OPT_0(A)) = O(n^\gamma)$ where γ is a constant to be determined later), otherwise $\mathcal{A}_{\text{approx}}$ is used. Note that the algorithm in Section 4.1 can be used for estimating *score*$(OPT_0(A))$.

First, we describe $\mathcal{A}_{\text{exact}}$. Recall that funny matrix multiplication for $N \times N$ integer matrices whose maximum absolute value of the entries is bounded by Q can be done in $O(Q(\log Q)N^\omega)$ time (Alon et al., 1991; Tamaki and Tokuyama, 1998). Using this in the algorithm in Section 3, we obtain $\mathcal{A}_{\text{exact}}$.

Lemma 1. *$\mathcal{A}_{exact}$ computes $OPT_0(A)$ in $O(Q(\log Q)n^\omega)$ time if score$(OPT_0(A)) \leq Q$.*

Proof: The maximum absolute value of entries in matrices appearing in the execution of $\mathcal{A}_{\text{exact}}$ is bounded by *score*$(OPT_0(A))$. Therefore, each funny matrix multiplication for $N \times N$ matrices can be done in $O(Q(\log Q)N^\omega)$ time. From Theorem 1, it is seen that the total time complexity is $O(Q(\log Q)N^\omega)$. □

Next, $\mathcal{A}_{\text{approx}}$ is obtained by modifying the original $O(n^3)$ time DP procedure $\mathcal{DP}_0$ for $\mathbf{RNA}_0$. Note that $S(i, j)$ in $\mathcal{DP}_0$ is equal to *score*$(OPT_0(a_i, a_{i+1}, \ldots, a_j))$.

Lemma 2. $|S(i, j) - S(i + h, j + k)| \leq |h| + |k|$.

Proof: From the definition of $\mathbf{RNA}_0$, both $|S(i, j) - S(i, j + 1)| \leq 1$ and $|S(i, j) - S(i + 1, j)| \leq 1$ hold. □

In $\mathcal{A}_{\text{approx}}$, we do not compute $\max_{i<k\leq j}\{S(i, k - 1) + S(k, j)\}$ exactly. Instead, we compute the maximum of $S(i, k-1)+S(k, j)$ for $O(n^\alpha + n^{1-\beta})$ values of k's (see figure 2), where α and $\beta (0 < \alpha, \beta < 1)$ are appropriate constants to be determined later.

We define a sequence of indices $f_i^+(h)$ and $f_j^-(h)$ for $h = 0, 1, 2, \ldots$ by

$$\begin{array}{ll} f_i^+(0) = i + \lceil n^\alpha \rceil, & f_j^-(0) = j - \lceil n^\alpha \rceil \\ f_i^+(h+1) = f_i^+(h) + \lceil (f_i^+(h) - i)^\beta \rceil, & f_j^-(h+1) = f_j^-(h) - \lceil (j - f_j^-(h))^\beta \rceil \end{array}$$

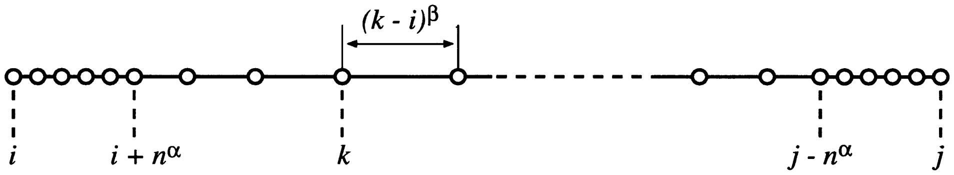

Figure 2. In $\mathcal{A}_{\text{approx}}$, $\max_k S(i, k - 1) + S(k, j)$ is computed not for all k, but for $O(n^\alpha + n^{1-\beta})$ values of k's, where such k's are represented by white circles in this figure.

Next, we define $\mathcal{I}(i, j)$ by

$$\mathcal{I}(i, j) = \{k \mid i < k \le n^{\alpha} \text{ or } j - n^{\alpha} \le k \le j\} \cup \{f_i^{+}(h) \mid f_i^{+}(h) \le (i + j)/2\} \\ \cup \{f_j^{-}(h) \mid f_j^{-}(h) \ge (i + j)/2\}.$$

Then, $\mathcal{A}_{\text{approx}}$ is expressed by the following DP procedure:

$$S'(i, j) = \max \begin{cases} S'(i + 1, j - 1) + \mu(a_i, a_j), \\ \max\limits_{k \in \mathcal{I}(i,j)} \{S'(i, k - 1) + S'(k, j)\}, \end{cases}$$

where we let $S'(i, j) = 0$ for $i \ge j$.

Lemma 3. *$\mathcal{A}_{approx}$ works in $O(n^{2+\alpha} + n^{3-\beta})$ time.*

Proof: Since $j - i \le n$, the size of $\mathcal{I}(i, j)$ is bounded by

$$|\mathcal{I}(i, j)| \le 2n^{\alpha} + 4\left(\frac{\frac{n}{2}}{\left(\frac{n}{2}\right)^{\beta}} + \frac{\frac{n}{4}}{\left(\frac{n}{4}\right)^{\beta}} + \frac{\frac{n}{8}}{\left(\frac{n}{8}\right)^{\beta}} + \cdots\right) \le O(n^{\alpha} + n^{1-\beta}).$$

Since $\max_{k \in \mathcal{I}(i,j)}\{S'(i, k - 1) + S'(k, j)\}$ is computed for $O(n^2)$ pairs (i, j), $\mathcal{A}_{\text{approx}}$ takes $O(n^{2+\alpha} + n^{3-\beta})$ time. □

Here, we define the *error* of an secondary structure M to $OPT_0(A)$ to be $score(OPT_0(A)) - score(M)$ (note that this value must be non-negative).

Lemma 4. *The error of a secondary structure M computed by $\mathcal{A}_{approx}$ is $O(n^{1+\alpha\beta-\alpha})$.*

Proof: Note that, for each (i, j), we define the *error* of $S'(i, j)$ (in $\mathcal{A}_{\text{approx}}$) to be $S(i, j) - S'(i, j)$. Here, we show that, for all i, j, the following inequality holds:

$$S(i, j) - S'(i, j) \le \max\{C \cdot m \cdot n^{\alpha\beta-\alpha} - C \cdot m^{\beta}, 0\}$$

for some constant C, where we let $m = j - i$. We prove this inequality by the induction on m.

Case (*i*) $m \le n^{\alpha}$. In this case, the error is always 0 and thus the inequality holds.
Case (*ii*) $m > n^{\alpha}$. In this case, we assume that the inequality holds for all m' such that $m' < m$, and we consider the following recurrence in $\mathcal{A}_{\text{approx}}$:

$$S'(i, j) = \max_{k \in \mathcal{I}(i,j)} \{S'(i, k - 1) + S'(k, j)\}.$$

Let k' be the integer maximizing $S'(i, k' - 1) + S'(k', j)$ under the condition that $i < k' \le j$, and let $k'' \in \mathcal{I}(i, j)$ be the integer maximizing $S'(i, k'' - 1) + S'(k'', j)$ under the condition that $k'' \in \mathcal{I}(i, j)$. From Lemma 2 and the definition of $\mathcal{I}(i, j)$, it is seen that

$S'(i, k'-1) + S'(k', j) - S'(i, k''-1) - S'(k'', j)$ is $O(h^\beta)$, where $h = \min(k''-1-i, j-k'')$. Then, the error of $S'(i, j)$ is bounded by

$$C \cdot h \cdot n^{\alpha\beta-\alpha} - C \cdot h^\beta + C \cdot (m-h) \cdot n^{\alpha\beta-\alpha} - C \cdot (m-h)^\beta + D \cdot h^\beta$$

where D is an appropriate constant, and we assume without loss of generality that $h > n^\alpha$. It is not difficult to verify that this value is at most $C \cdot m \cdot n^{\alpha\beta-\alpha} - C \cdot m^\beta$ for $C \gg D$. □

Theorem 5. *For* $\mathbf{RNA}_0$, *an RNA secondary structure with the score at least* $1 - \epsilon$ *of the maximum can be computed in* $O(n^{2.776} + (1/\epsilon)^{O(1)})$ *time*, *where* ϵ *is any positive constant number.*

Proof: First, we estimate *score* $(OPT_0(A))$ using the algorithm described in Section 4.1. If the estimated value is at most n^γ, an optimal structure is computed using $\mathcal{A}_{\text{exact}}$, otherwise an approximate structure is computed using $\mathcal{A}_{\text{approx}}$. Then, from Lemma 3, the time complexity is $O(n^{\gamma+\omega} \log n + n^{2+\alpha} + n^{3-\beta})$.

From Lemma 4, the ratio of the score of an approximate solution computed by $\mathcal{A}_{\text{approx}}$ to the optimal score is $(score(OPT_0(A)) - O(n^{1+\alpha\beta-\alpha}))/score(OPT_0(A))$, which is greater than $1 - \epsilon$ for any constant $\epsilon > 0$ if $1 + \alpha\beta - \alpha < \gamma$ and $n > (1/\epsilon)^c$ hold where c is a constant depending on α, β and γ. Note that we can compute an optimal solution using $O((1/\epsilon)^{3c})$ time if $n \le (1/\epsilon)^c$.

Here, we let $\alpha = 0.776$, $\beta = 0.224$, $\gamma = 0.398$ and $\omega = 2.376$. Then, $1 + \alpha\beta - \alpha < \gamma$ is satisfied and the theorem follows. □

5. Approximation algorithms for more practical cases

Although the above algorithms are not practical, the technique developed for $\mathcal{A}_{\text{approx}}$ can be applied to more practical versions of RNA secondary structure prediction. Since the quality of a predicted RNA structure heavily depends on the energy function, many practical versions have been proposed based on various energy functions (Setubal and Meidanis, 1997; Turner et al., 1988; Waterman, 1995). In this section, we show that the developed technique can be applied to some of them.

5.1. *Energy function for adjacent base pairs*

In $\mathbf{RNA}_0$, energy function is defined for each base pair. On the other hand, energy functions defined for adjacent base pairs are widely used (Turner et al., 1988; Uemura et al., 1995). In this case, energy function μ is defined for adjacent base pairs $(a_i a_{i+1}, a_{j-1} a_j)$. Formally, an energy function is a function from $\Sigma \times \Sigma \times \Sigma \times \Sigma$ to the set of *negative* reals. Note that, in this case, the global free-energy (i.e., the total score) is always negative and the problem is defined as a minimization problem.

Under this kind of energy functions, an optimal RNA secondary structure can be computed in $O(n^3)$ time using a DP procedure similar to $\mathcal{DP}_0$ (Turner et al., 1988). Moreover, a context-free grammar (with score) can also be associated as in Section 2, and thus we can obtain an $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$ time exact algorithm.

Since energy function $\mu(a_i a_{i+1}, a_{j-1} a_j)$ takes values between 0 and E where E is a negative constant, the property similar to Lemma 2 still holds in this case, and thus we can obtain an approximation algorithm as in Section 4.

Theorem 6. *Under an energy function defined for adjacent base pairs, an optimal RNA secondary structure can be computed in* $O(n^3(\log\log n)^{1/2}/(\log n)^{1/2})$ *time, and an RNA secondary structure with the free-energy at most* $1-\epsilon$ *of the minimum can be computed in* $O(n^{2.776}+(1/\epsilon)^{O(1)})$ *time, where* ϵ *is any positive constant number.*

5.2. Destabilizing energy

We did not consider free-energy for unpaired residues so far. However, such residues are also important determinants of RNA stability, and several energy functions are proposed for unpaired residues (Setubal and Meidanis, 1997; Uemura et al., 1995; Waterman, 1995). A maximal consecutive part of unpaired residues is called a *loop*, where there are several kinds of loops such as *bulge loop*, *end loop* and *interior loop* (Waterman, 1995). Usually, an energy function for loops takes positive value and is called a *destabilizing* energy function.

Waterman and Smith (1986) proposed an $O(n^3)$ time algorithm for computing locally destabilizing RNA secondary structures (i.e., minimum energy RNA secondary structures without *multibranch loops*). They consider destabilizing energies for *loop*, *bulge* and *interior loop*. In their algorithm, the optimal (minimum) score $S_{i,j}$ for subsequence $a_i,\ldots,a_j$ is computed by taking the minimum of the following scores (free-energies):

(a) $\mu(a_i,a_j)+\xi(j-i-1)$,
(b) $\mu(a_i,a_j)+\eta+S_{i+1,j-1}$,
(c) $\min_{k\geq 1}\{\mu(a_i,a_j)+\nu(k)+S_{i+k+1,j-1}\}$,
(d) $\min_{k\geq 1}\{\mu(a_i,a_j)+\nu(k)+S_{i+1,j-k-1}\}$,
(e) $\min_{k_1,k_2\geq 1}\{\mu(a_i,a_j)+\gamma(k_1+k_2)+S_{i+1+k_1,j-1-k_2}\}$,

where μ,ξ,η,ν,γ are energy functions (note that symbols $S_{i,j}$, μ, ν are used here, instead of $h_{i,j}$, α, β in (Waterman and Smith, 1995).

Although it takes $O(n^4)$ time if we directly compute the above recurrence, it can be reduced to $O(n^3)$ using a table $S^*_{i,j}(k)$ defined by

$$S^*_{i,j}(k)=\min\{S_{i',j'} \mid (j-j')+(i'-i)-2=k,\, i'\geq i+2,\, j'\leq j-2\}.$$

In this case, the energy corresponding to (e) is computed by

$$\min\{\mu(a_i,a_j)+\gamma(k+2)+S^*_{i,j}(k)\}.$$

Since $S^*_{i,j}(k)$ is computed in $O(1)$ time from $S^*_{i,j-1}(k-1)$ by

$$S^*_{i,j}(k)=\min\{S^*_{i,j-1}(k-1),\, S_{i+k,j-2}\},$$

the total time complexity is $O(n^3)$.

Although we do not yet succeed to develop an improved exact algorithm, we can develop an $O(n^3)$ time approximation algorithm for a special case.

In order to develop an approximation algorithm, we first modify (c) and (d), where (c) and (d) correspond to free-energies for *bulge loops*. In the original algorithm, 'min' in (c) and (d) is computed for $O(n^2)$ pairs of (i, j), and thus the computation time for bulge loops is $O(n^3)$. However, we can reduce the computation time for bulge loops to $O(n^{2+\alpha} + n^{3-\beta})$ if we compute 'min' only for k's such that $i + k \in \mathcal{I}(i, j)$ as in Section 4.2.

In order to reduce the computation time for (e), we compute approximate value $\tilde{S}^*_{i,j}(k)$ of $S^*_{i,j}(k)$ only for k's in

$$\{1, 2, 3, \ldots, \lceil n^\alpha \rceil\} \cup \{(j \bmod L) + f_i^+(h)\},$$

where $L = \lceil n^{\alpha\beta} \rceil$. If $j \leq \lceil n^\alpha \rceil$ or $(j \bmod L) \neq 0$, $\tilde{S}^*_{i,j}(k)$ is computed in $O(1)$ time as in the original procedure. Otherwise, $\tilde{S}^*_{i,j}(k)$ is computed by

$$\begin{aligned}\tilde{S}^*_{i,j}(k) = \min\{\tilde{S}_{i',j'} \mid (j - j') + (i' - i) - 2 = k, (j' \bmod L) = 0,\\ i' \geq i + 2, j' \leq j - 2\},\end{aligned}$$

where $S_{i,j}$ is replaced by approximate score $\tilde{S}_{i,j}$ as in Section 4. Then, the total size for table $\tilde{S}^*_{i,j}(k)$ is $O(n^2 \cdot (n^\alpha + n^{1-\beta})) = O(n^{2+\alpha} + n^{3-\beta})$ and the total time for computing this table is

$$O\left(n^{2+\alpha} + n^{3-\beta} + n \cdot \frac{n}{n^{\alpha\beta}} \cdot (n^{1-\beta}) \cdot \frac{n}{n^{\alpha\beta}}\right) = O(n^{2+\alpha} + n^{3-\beta} + n^{4-\beta-2\alpha\beta}).$$

Therefore, the total time complexity of the approximation algorithm is $O(n^{2+\alpha} + n^{3-\beta} + n^{4-\beta-2\alpha\beta})$.

In order to analyze the error due to this algorithm, we require a reasonable assumption that

$$|\nu(k) - \nu(k+1)| \leq B, \quad |\gamma(k) - \gamma(k+1)| \leq B$$

hold for all k, where B is a some constant. We call energy functions satisfying this assumption *smooth* energy functions. Since the values of most energy functions change gradually as n grows (Turner et al., 1988; Waterman, 1995), most energy functions can be considered as smooth functions. For example, all of linear destabilizing functions are smooth functions. Under the assumption of smooth energy functions, we analyze error due to each of (a)–(e). For (a) and (b), there is no error. The error due to one execution of (c) or (d) is $O(B \cdot k^\beta)$ if $k > \lceil n^\alpha \rceil$, otherwise it is 0. The error due to one execution of (e) is $O(B \cdot k^\beta)$ if $k > \lceil n^\alpha \rceil$, otherwise it is 0. Therefore, the analysis in Lemma 4 can also be applied to this case.

Theorem 7. *Under a smooth energy function including destabilizing energy, an RNA secondary structure without multibranch loops whose free-energy is at most $O(B \cdot n^{1+\alpha\beta-\alpha})$ larger than the minimum can be computed in $O(n^{2+\alpha} + n^{3-\beta} + n^{4-\beta-2\alpha\beta})$ time.*

Letting $\alpha = 4/5$ and $\beta = 1/2$, we have:

Corollary 2. *Under a smooth energy function including destabilizing energy, an RNA secondary structure without multibranch loops whose free-energy* (<0) *is at most* $1-\epsilon$ *of the minimum* (<0) *can be computed in* $O(n^{2.8}+(B/(C\epsilon))^{O(1/\delta)})$ *time if the absolute value of the minimum free-energy is at least* $C\cdot n^{0.6+\delta}$ *for some constant* C, *where* $\epsilon,\delta>0$ *are arbitrarily small constants.*

5.3. Pseudoknots

Although *pseudoknots* (special kinds of substructures) are taken into account in a few algorithms, pseudoknots appear in several important RNAs (Uemura et al., 1995). For a basic version (i.e., maximizing the number of base pairs) of RNA secondary structure prediction with *simple* pseudoknots, an $O(n^4)$ time algorithm was proposed by Uemura et al. (1995) based on *tree-adjoining* grammar, and then a simpler $O(n^4)$ time algorithm was developed without tree-adjoining grammar (Akutsu, 1997). We can apply a similar technique to the latter algorithm and obtain an $O(n^{3.5})$ time algorithm with error at most $O(n^{0.75})$. Details will be described in a journal version of the latter paper (Akutsu, 1997).

6. Approximation algorithm for SCFG

Since there is a close relationship between $\mathbf{RNA}_0$ and SCFG, it is natural to try to extend Theorem 5 for SCFG. Unfortunately, Lemma 2 or similar property does not hold for general SCFG and thus we can not directly extend Theorem 5. However, we can still compute approximate scores for SCFG in $O(n^{3-\delta})$ time for some δ by modifying Valiant's algorithm.

Since we consider a fixed grammar, we assume that the score associated with each production rule is bounded by a constant. Moreover, we assume that each score is represented with finite bits. Then, we can assume that the optimal score is $O(n)$.

In Section 3, we used funny matrix multiplication for computing an optimal parse tree. In this section, we use the following approximate funny matrix multiplication, where Tamaki and Tokuyama (1998) used a similar technique.

Lemma 5. *Let* A, B *be* $N\times N$ *integer matrices and let* C *be a funny matrix product of* A *and* $B(C=A\odot B)$. *If the absolute value of each entry of* A *is bounded by* $K\cdot N$ *for some constant* K, *we can compute in* $O(N^{\omega+0.5}\log N)$ *time a matrix* $C'=(c'_{ij})$ *such that error of each entry (i.e.,* $|c'_{ij}-c_{ij}|$*) is* $O(\sqrt{N})$.

Proof: Without loss of generality, we assume all entries of A, B are non-negative.

Let $A'=(a'_{ij})$ be a matrix defined by

$$a'_{ij}=\lfloor a_{ij}/R\rfloor,$$

where $R=\lceil\sqrt{N}\rceil$. Let $B'=(b'_{ij})$ be a matrix defined by

$$b'_{ij}=\begin{cases}\lfloor(b_{ij}-b_j)/R\rfloor, & \text{if } b_{ij}\geq b_j,\\ 0, & \text{otherwise,}\end{cases}$$

where

$$b_j = \left(\left\lfloor \frac{(\max_i b_{ij})}{KN} \right\rfloor - 2 \right) \cdot KN$$

(if this value is negative, we let $b_j = 0$).

We compute $C' = R^2 \cdot (A' \odot B') + B''$ using $O(N^{\omega+0.5} \log N)$ time, where $B'' = (b''_{ij})$ is defined by $b''_{ij} = b_j$. Then, c'_{ij} satisfies $|c'_{ij} - c_{ij}| \leq 2R$. □

Theorem 8. *A parse tree for SCFG whose score is at least* $1 - \epsilon$ *of the maximum can be computed in* $O(n^{2.976} \log n + (1/\epsilon)^{O(1)})$ *time, where* ϵ *is any positive constant number.*

Proof: As in Theorem 5, we combine an approximation algorithm and an exact algorithm. In the approximation algorithm, we use a modified Valiant's algorithm in which boolean matrix multiplication is replaced by approximate funny matrix multiplication.

Recall that x_{ij} denotes the score of a parse tree for subsequence $s_i, s_{i+1}, \ldots, s_{j-1}$ in Theorem 3. Here, we compute the exact score for an optimal parse tree for each subsequence with length less than n^α and we compute an approximate score for each subsequence with length at least n^α. Moreover, if s' is obtained by concatenating s^1 and s^2 (using some production rule), we will make the error due to this concatenation be at most $O(\sqrt{\min\{|s^1|, |s^2|\}}$ (we will make the error be 0 if $\min\{|s^1|, |s^2|\} < n^\alpha$), where $|x|$ denotes the length of sequence x. If we can do so, the total error is $O(n^{1+\alpha\beta-\alpha})$ as in Lemma 4. Note that we let $\alpha = 4/5$, $\beta = 1/2$ in this proof, and thus the total error is $O(n^{3/5})$.

Carefully checking (modified) Valiant's algorithm, we can see that, in each funny matrix multiplication for $N \times N$ matrices, at least one of input matrices P has the following property: each entry of P represents the score of an optimal parse tree for subsequence of length at most $2N$. Using this property, we compute approximate funny matrix product for P, Q in the following way.

First we assume P (resp. Q) satisfies the above property and each entry of Q (resp. P) corresponds to a sequence with length at least N. We divide P (resp. Q) into four $N/2 \times N/2$ submatrices. Then, we have

$$\begin{aligned} P \odot Q &= \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix} \odot \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix} \\ &= \begin{pmatrix} P_1 \odot Q_1 + P_2 \odot Q_3 & P_1 \odot Q_2 + P_2 \odot Q_4 \\ P_3 \odot Q_1 + P_4 \odot Q_3 & P_3 \odot Q_2 + P_4 \odot Q_4 \end{pmatrix} \end{aligned}$$

where '$+$' means 'max' in this case. If $N < n^\alpha$, we compute funny matrix product $P \odot Q$ exactly using $O(N^3)$ time. If $N \geq n^\alpha$, we compute funny matrix product approximately using the following recursive procedure. Since each entry in P_1, P_2, P_4 corresponds to a substring of length at least $N/2$, we compute approximate funny matrix products using Lemma 5 instead of $P_i \odot Q_j$ for $i \neq 3$. Since each entry in P_3 corresponds to a substring of length at most N, we apply the same procedure recursively to the approximate computation of $P_3 \odot Q_1$ and $P_3 \odot Q_2$. Then, from Lemma 5, the error due to the concatenation is at most $O(\sqrt{\min\{|s^1|, |s^2|\}})(=O(\sqrt{|s^1|}))$ for any sequences s^1 and s^2 corresponding to some entries in A and B respectively. The time $M(N)$ for this approximate funny matrix multiplication is

$$M(N) = \begin{cases} 6 \cdot O((N/2)^{\omega+0.5} \log N) + 2 \cdot M(N/2) + O(N^2), & \text{if } N \geq n^{\alpha}, \\ O(N^3), & \text{otherwise.} \end{cases}$$

Since $2^k \cdot (n^{\alpha})^3 = N \cdot n^{2\alpha}$ holds for k satisfying $N/2^k = n^{\alpha}$, we have

$$M(N) = \begin{cases} O(N^{\omega+0.5} \log N + Nn^{2\alpha}), & \text{if } N \geq n^{\alpha}, \\ O(N^3), & \text{otherwise.} \end{cases}$$

Next we consider the other case: both P and Q satisfy the property. In this case, we must compute $P_3 \odot Q_1$, $P_3 \odot Q_2$, $P_2 \odot Q_3$ and $P_4 \odot Q_3$ recursively. Then, $M(N)$ should satisfy:

$$M(N) = \begin{cases} 4 \cdot O((N/2)^{\omega+0.5} \log N) + 4 \cdot M(N/2) + O(N^2), & \text{if } N \geq n^{\alpha}, \\ O(N^3), & \text{otherwise.} \end{cases}$$

Since $4^k \cdot (n^{\alpha})^3 = N^2 \cdot n^{\alpha}$ holds for k satisfying $N/2^k = n^{\alpha}$, we have

$$M(N) = \begin{cases} O(N^{\omega+0.5} \log N + N^2 n^{\alpha}), & \text{if } N \geq n^{\alpha}, \\ O(N^3), & \text{otherwise.} \end{cases}$$

Therefore, $M(N)$ is bounded by $O(\min(N^{\omega+0.5} \log N + N^2 n^{\alpha}, N^3))$. Since this still satisfies the condition of Theorem 1, the time complexity for computing an approximate parse tree is $O(n^{\omega+0.5} \log n)$.

On the other hand, we compute an optimal parse tree simultaneously (assuming the score of an optimal parse tree is $O(n^{3/5})$) using $O(n^{\omega+\frac{3}{5}} \log n)$ time. Since $n^{\omega+0.5} < n^{\omega+\frac{3}{5}}$, the total computation time is $O(n^{\omega+\frac{3}{5}} \log n)$.

Since $O((1/\epsilon)^{O(1)})$ time is required for small n as in Theorem 5, the total time complexity is $O(n^{\omega+\frac{3}{5}} \log n + (1/\epsilon)^{O(1)})$. □

Note that if we consider the probability, the ratio of the probability of an approximate parse tree to the probability of an optimal parse tree is bounded by p^{ϵ} for any $\epsilon > 1$ where p is the probability of an optimal parse tree.

7. Concluding remarks

In this paper, we proposed approximation and exact algorithms for RNA secondary structure prediction and SCFG. The most important contribution of this paper is that it shows that the well-known $O(n^3)$ time DP algorithms are not necessarily optimal.

Although the exact algorithms are complicated, the approximation algorithms (excluding $\mathcal{A}_{\text{exact}}$) are very simple and might be practical. Of course, secondary structures obtained by the approximation algorithms may be different from optimal secondary structures. However, optimal secondary structures do not necessarily coincide with real secondary structures because empirically derived energy functions are used in the computation of optimal secondary structures, and thus optimal secondary structures are also approximations of real secondary structures. Indeed, many heuristic algorithms without guaranteed approximation ratio have been proposed for RNA secondary structure prediction (Abrahams et al., 1990). Therefore, the proposed approximation algorithms may be practical.

Finally, we conclude with two open problems:

(i) Development of an $O(n^{3-\delta})$ time exact algorithm for $\mathbf{RNA}_0$ and/or SCFG, where δ is some positive constant,

(ii) Removement of the assumption on the absolute value of free-energy in Corollary 2.

References

J.P. Abrahams, M. Berg, E. Batenburg, and C. Pleij, "Prediction of RNA secondary structure, including pseudoknotting by computer simulation," *Nucleic Acids Research*, vol. 18, pp. 3035–3044, 1990.

T. Akutsu, "DP algorithms for RNA secondary structure prediction with pseudoknots," *Genome Informatics 1997*, Universal Academy Press: Tokyo, 1997, pp. 173–179.

N. Alon, Z. Galil, and O. Margalit, "On the exponent of the all pairs shortest path problem," in *Proc. 32nd IEEE Symp. Foundations of Computer Science*, IEEE, 1991, pp. 569–575.

D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progression," *J. Symbolic Computation*, vol. 9, pp. 251–280, 1990.

D. Eppstein, Z. Galil, and R. Giancarlo, "Speeding up dynamic programming," in *Proc. 29th IEEE Symp. Foundations of Computer Science*, IEEE, 1988, pp. 488–496.

D. Eppstein, Z. Galil, R. Giancarlo, and G.F. Italiano, "Sparse dynamic programming II: Convex and concave cost functions," *J. ACM*, vol. 39, pp. 546–567, 1992.

M.L. Fredman, "New bounds on the complexity of the shortest path problem," *SIAM Journal on Computing*, vol. 5, pp. 83–89, 1976.

Z. Galil and K. Park, "Dynamic programming with convexity, concavity and sparsity," *Theoretical Computer Science*, vol. 92, pp. 49–76, 1992.

M. Kanehisa and W.B. Goad, "Pattern recognition in nucleic acid sequences II: An efficient method for finding locally stable secondary structures," *Nucleic Acids Research*, vol. 10, pp. 265–277, 1982.

L.L. Larmore and B. Schieber, "On-line dynamic programming with applications to the prediction of RNA secondary structure," *Journal of Algorithms*, vol. 12, pp. 490–515, 1991.

Y. Sakakibara, M. Brown, E. Hughey, I.S. Mian, K. Sjlölander, R.C. Underwood, and D. Haussler, "Stochastic context-free grammars for tRNA modeling," *Nucleic Acids Research*, vol. 22, pp. 5112–5120, 1994.

J. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*, PWS Pub. Co.: Boston, 1997.

T. Takaoka, "A new upper bound on the complexity of all pairs shortest path problem," *Information Processing Letters*, vol. 43, pp. 195–199, 1992.

H. Tamaki and T. Tokuyama, "Algorithms for maximum subarray problem based on matrix multiplication," in *Proc. 9th ACM-SIAM Symp. Discrete Algorithms*, ACM, 1998, pp. 446–452.

D.H. Turner, N. Sugimoto, and S.M. Freier, "RNA structure prediction," *Ann. Rev. Biophys. Chem.*, vol. 17, pp. 167–192, 1988.

Y. Uemura, A. Hasegawa, S. Kobayashi, and T. Yokomori, "Grammatically modeling and predicting RNA secondary structures," in *Proc. Genome Informatics Workshop VI*, Universal Academy Press: Tokyo, 1995, pp. 67–76.

L.G. Valiant, "General context-free recognition in less than cubic time," *Journal of Computer and System Sciences*, vol. 10, pp. 308–315, 1975.

M.S. Waterman and T.F. Smith, "RNA secondary structure: A complete mathematical analysis," *Math. Biosciences*, vol. 41, pp. 257–266, 1978.

M.S. Waterman and T.F. Smith, "Rapid dynamic programming algorithms for RNA secondary structure," *Advances in Applied Mathematics*, vol. 7, pp. 455–464, 1986.

M.S. Waterman, *Introduction to Computational Biology*, Chapman & Hall: London, 1995.

M. Zuker and P. Stiegler, "Optimal computer folding for large RNA sequences using thermodynamics and auxiliary information," *Nucleic Acids Research*, vol. 9, pp. 133–148, 1981.