PublisherInfo		
PublisherName	:	Springer International Publishing
PublisherLocation	:	Cham
PublisherImprintName	:	Springer

Efficient Alignment of RNAs with Pseudoknots Using Sequence Alignment Constraints

ArticleInfo		
ArticleID	:	114
ArticleDOI	:	10.1155/2009/491074
ArticleCitationID	:	491074
ArticleSequenceNumber	:	13
ArticleCategory	:	Research Article
ArticleCollection	:	Applications of Signal Processing Techniques to Bioinformatics, Genomics, and Proteomics
ArticleFirstPage	:	1
ArticleLastPage	:	1
ArticleHistory	:	RegistrationDate : 2008–6–18 Received : 2008–6–18 Revised : 2008–11–13 Accepted : 2009–1–16 OnlineDate : 2009–2–12
ArticleCopyright	:	Byung-Jun Yoon.2009 This article is published under license to BioMed Central Ltd. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
ArticleGrants	:	
ArticleContext	:	136372009200911

Byung-Jun Yoon,[Aff1]
Corresponding Affiliation: Aff1
Email: bjyoon@ece.tamu.edu

Aff1 Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA

Abstract

When aligning RNAs, it is important to consider both the secondary structure similarity and primary sequence similarity to find an accurate alignment. However, algorithms that can handle RNA secondary structures typically have high computational complexity that limits their utility. For this reason, there have been a number of attempts to find useful alignment constraints that can reduce the computations without sacrificing the alignment accuracy. In this paper, we propose a new method for finding effective alignment constraints for fast and accurate structural alignment of RNAs, including pseudoknots. In the proposed method, we use a profile-HMM to identify the â€œseedâ€ regions that can be aligned with high confidence. We also estimate the position range of the aligned bases that are located outside the seed regions. The location of the seed regions and the estimated range of the alignment positions are then used to establish the sequence alignment constraints. We incorporated the proposed constraints into the profile context-sensitive HMM (profile-csHMM) based RNA structural alignment algorithm. Experiments indicate that the proposed method can make the alignment speed up to 11 times faster without degrading the accuracy of the RNA alignment.

1. Introduction

Sequence alignment lies at the heart of various computational methods that are used for analyzing biological sequences, such as RNAs and proteins. Alignment algorithms have been extensively used for comparing sequences to identify homologues, predict their structures, and infer their biological functions. Many functional noncoding RNAs (ncRNAs) are known to conserve their base paired secondary structure as well as their primary sequence [1]. For this reason, when aligning RNAs, it is important to consider both structure and sequence similarities in order to find an accurate alignment that is biologically meaningful. For a similar reason, it is expedient to employ scoring schemes that can reasonably combine contributions from the secondary structure similarity as well as primary sequence similarity when performing an RNA similarity search, as it can significantly reduce the number of false-positive predictions [2].

Conservation of the secondary structure gives rise to complicated symbol correlations between the pairing bases in the RNA sequence. Therefore, in order to take structural similarity into account, RNA alignment and search algorithms need to handle these base correlations in a principled manner. Until now, a number of probabilistic models have been proposed for this purpose [2, 3], where *stochastic context-free grammars (SCFGs)* and their variants have been especially popular. A typical problem of these models and the relevant algorithms is the high computational complexity. For example, the *Cocke-Younger-Kasami (CYK) algorithm* used in the SCFG-based alignment and search has a complexity of $O(L^3)$, where L is the length of the RNA that is to be aligned. Algorithms for simultaneous folding and alignment of RNAs (typically referred to as the *Sankoff algorithm*) have an even higher complexity, which require $O(L^{3N})$ computations for aligning N RNAs of length L [4]. The aforementioned algorithms do not consider pseudoknots, which are RNA secondary structures with crossing base-pairs, and there will be a steep increase in complexity if we begin to consider such pseudoknots. Pseudoknots are often ignored by many algorithms since they significantly increase the computational complexity.

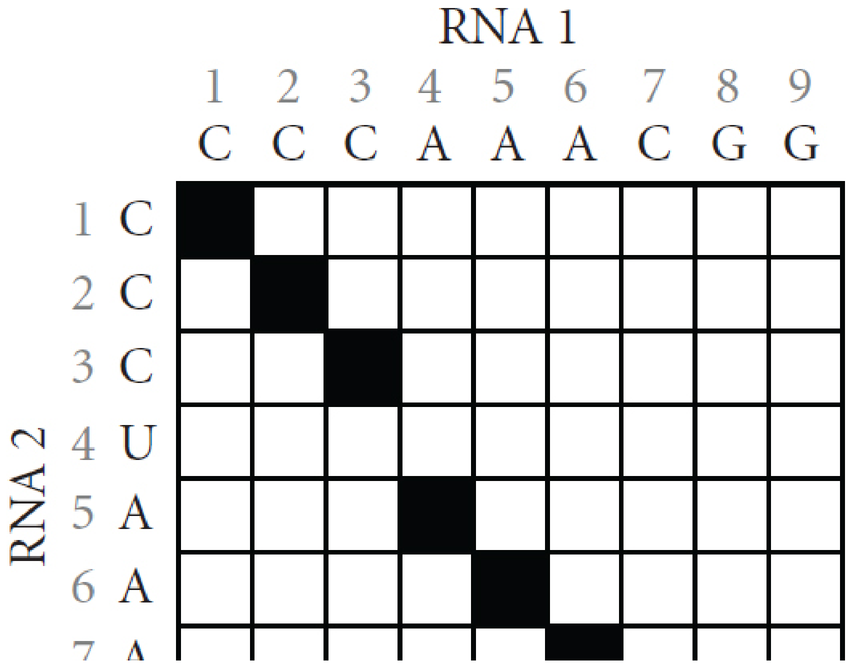
The high computational cost of RNA alignment and search algorithms limits their utility in practical applications, especially when the RNA of interest is long. To cope with this problem, there have been extensive research efforts to develop heuristic methods that can make these algorithms faster without degrading the accuracy. For example, let us consider the simultaneous folding and alignment algorithm. Its computational complexity is already $O(L^6)$ for aligning just two RNAs, making it practically unusable for a larger number of RNAs. Even for pairwise alignments, the algorithm becomes quickly infeasible as the RNAs get longer. Therefore, in order to utilize these algorithms in practical applications, it is essential that we first reduce their computations. For this reason, most of the pairwise RNA alignment algorithms adopt various tricks to minimize the alignment time [5–11]. Similarly, a number of methods have been proposed to make RNA similarity searching faster, where the prescreening approach is a good example [12–15]. The prescreening approach uses a simple model, such as a *profile hidden Markov model (profile-HMM)*, to identify the regions that have a reasonable amount of (sequence) similarity. Only these regions will be passed to a more complex model, such as a *covariance model (CM; profile-SCFG)* [3] or a *profile context-sensitive HMM (profile-csHMM)* [16, 17], for further inspection. In fact, these are just a few examples, and there also exist other approaches for making RNA alignment and RNA search algorithms faster [18, 19].

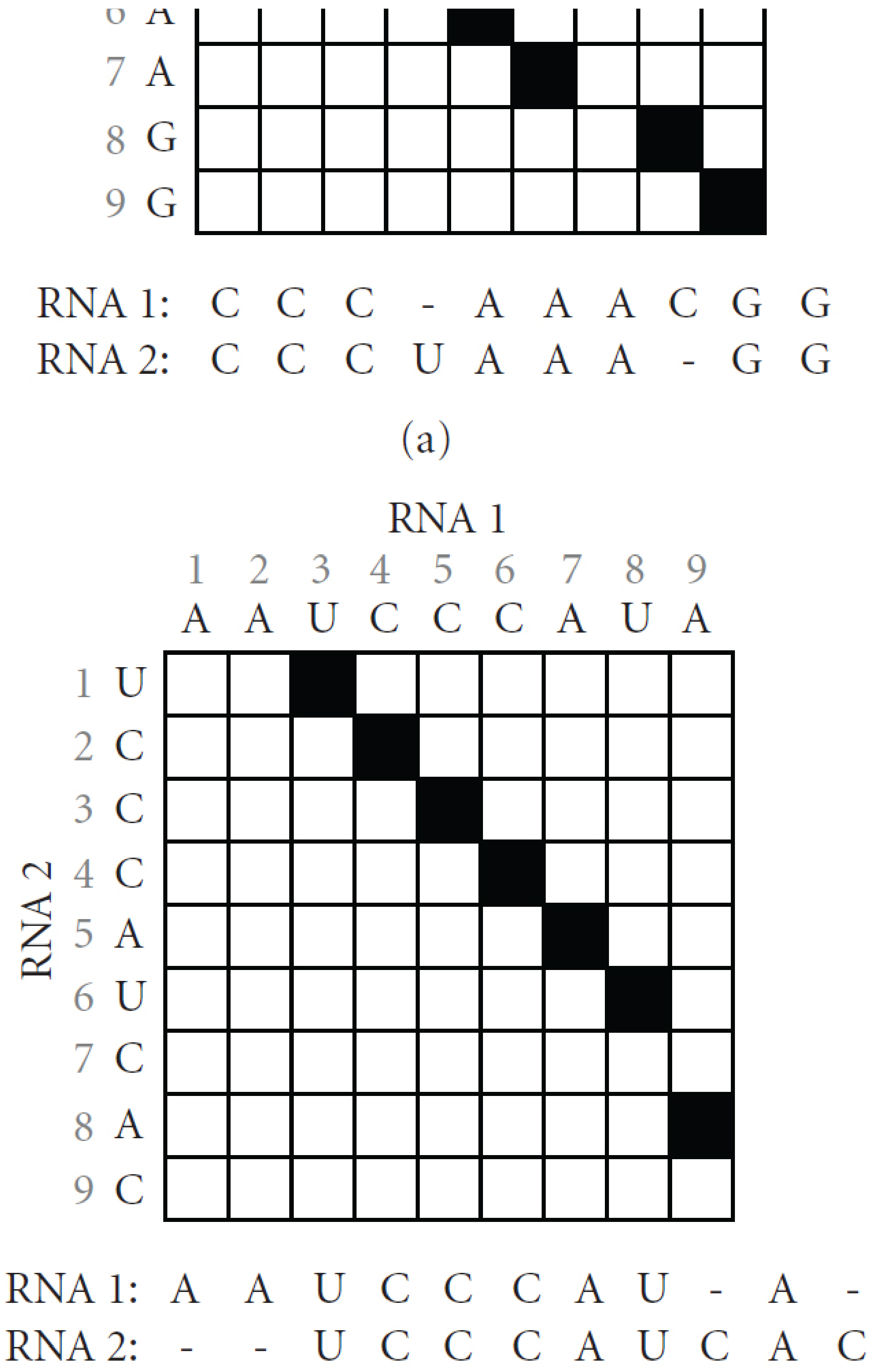
Recently, we proposed an efficient RNA structural alignment algorithm based on profile-csHMMs, which can also be used for aligning RNAs that contain pseudoknots [17]. This algorithm finds the optimal alignment between a structured reference RNA and an unstructured target RNA, by taking both structure and sequence similarities into account. It was demonstrated that the profile-csHMM algorithm can find accurate alignment of RNA pseudoknots [17]. In this paper, we propose a novel method for finding effective sequence alignment constraints that can improve the computational efficiency of the profile-csHMM structural alignment algorithm. The overall organization of the paper is as follows. In Section 2, we describe the concept of constrained alignment and briefly review some of the existing methods for finding the alignment constraints. After the review, we propose a new method for estimating the alignment constraints in Section 3. Finally, Section 4 describes how these constraints can be incorporated into the profile-csHMM based structural alignment method. Experimental results will be presented at the end of Section 4, which demonstrate the effectiveness of the proposed approach.

2. Constrained Sequence Alignment

Let us assume that we want to find the alignment of two RNAs $\mathbf{X} = x_1x_2\cdots x_{L_1}$ (RNA-1) and $\mathbf{Y} = y_1y_2\cdots y_{L_2}$ (RNA-2). The predicted sequence alignment can be uniquely represented by the set of aligned bases (x_m,y_n) in a matrix. For example, let us consider the RNA alignment in Figure 1(a). The matrix shows the positions of the aligned bases, where a black square at (m,n) indicates that the bases x_m and y_n are aligned to each other. As we can see in Figure 1(a), the sequence alignment can be represented by a "path" of aligned base positions (m,n) in the alignment matrix. Another example is shown in Figure 1(b) for a slightly different alignment. Without any prior constraints, any base in one RNA can be aligned to any base in the other RNA, hence the "alignment path" can be located anywhere in the matrix.

Figure 1 (a), (b) Examples of sequence alignment. The black squares denote the aligned positions. (c) Illustration of constrained alignment. The final sequence alignment (shown in black) should lie inside the constrained region (shown in gray).



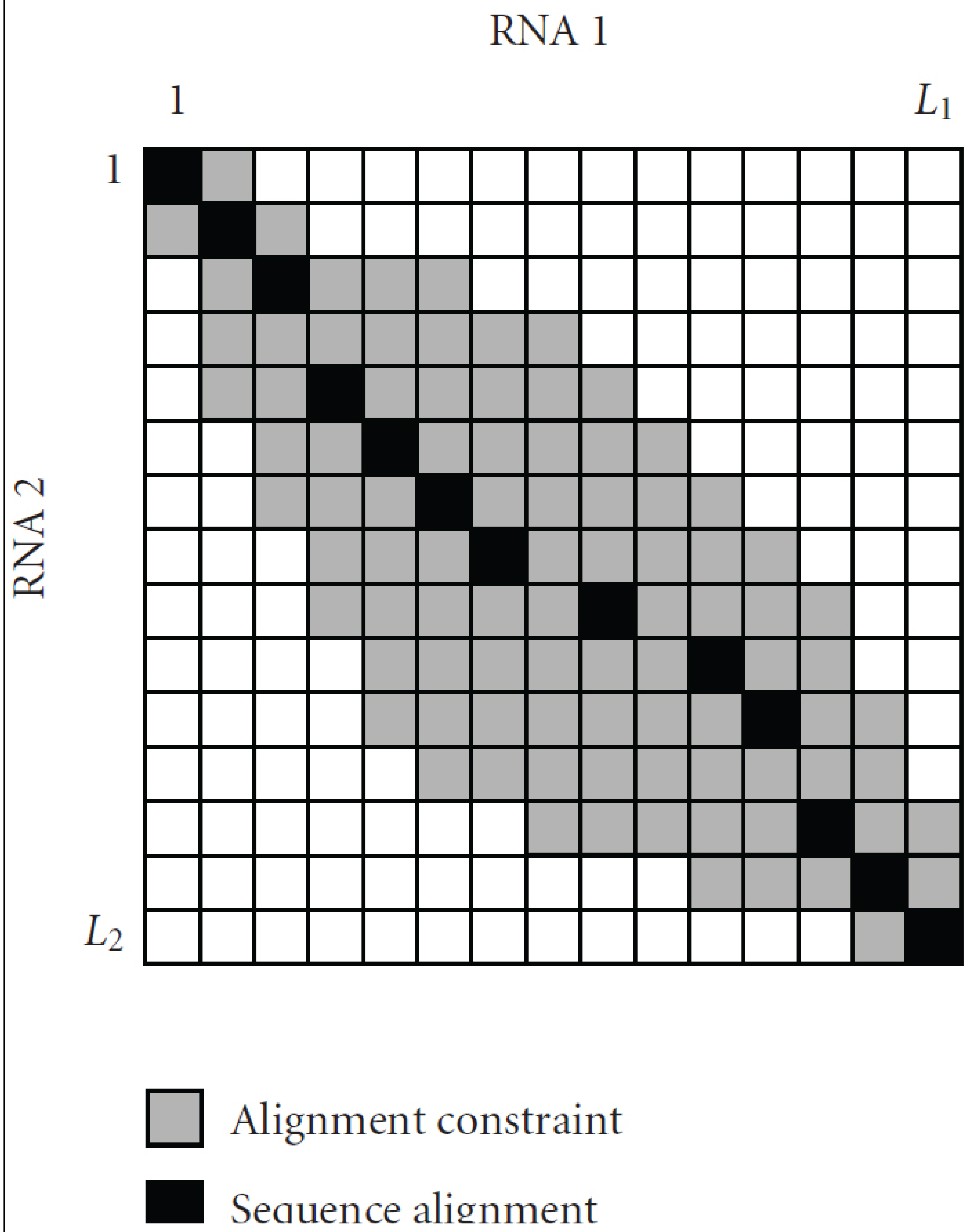


RNA 1: C C C - A A A C G G
RNA 2: C C C U A A A - G G

(a)

RNA 1: A A U C C C A U - A -
RNA 2: - - U C C C A U C A C

(b)


RNA 1
1
L_1
1
RNA 2
L_2
Alignment constraint
Sequence alignment

(c)

Now, assume that we are given some prior information about the region where the aligned bases should be located. This is illustrated in Figure 1(c), where the gray area depicts the possible position (m,n) of the aligned bases x_m and y_n. Knowing that all the aligned bases should be located inside the given region, we only have to consider the alignment paths that are contained in this region when finding the RNA alignment. Consequently, the search space for finding the optimal alignment is reduced, bringing down the overall computational cost. The reduction in time achieved by restricting the alignment space is typically much larger compared to the reduction in the alignment space itself, since the computational complexity of most RNA alignment algorithms is a high-order polynomial of the RNA length. It has to be noted that the alignment accuracy will not be affected as far as the optimal alignment (the black path in Figure 1(c)) is contained in the constraining region (the gray area in Figure 1(c)).

As illustrated in the previous example, using appropriate sequence alignment constraints can greatly enhance the efficiency of the alignment algorithm. So, the natural question is how we can predict good alignment constraints that can minimize the alignment time without degrading the alignment accuracy. Until now, various methods have been proposed for restricting the alignment space to improve the efficiency of diverse RNA alignment and search algorithms [5–9, 11, 17, 18]. For example, the query-dependent banding (QDB) method [18] is used to make CM-based RNA alignment algorithms faster, by excluding the regions in the dynamic programming matrix that have insignificant probability. These regions can be precomputed based on the given CM and do not depend on the target database. *Foldalign* [7], an algorithm for simultaneous RNA folding and alignment, limits the maximum length of the RNA-motif as well as the maximum length difference between the subsequences that are being compared. Recent implementation of Foldalign [8] adopts a heuristic that prunes the dynamic programming matrix in order to reduce the overall time and memory requirements. Another RNA alignment and structure prediction algorithm called *Stemloc* [9] constrains the solution space by using "fold envelopes" and "alignment envelopes." The fold envelopes are used to restrict the search over secondary structures and the alignment envelopes are used to restrict the possible alignments between the given sequences.

A recent implementation of *Dynalign* [11], a joint alignment and secondary structure prediction algorithm for two RNAs, assumes that the aligned bases in the respective RNAs should be located within a certain distance. To be more precise, the mth base x_m in RNA-1 ($\mathbf{X} = x_1x_2\cdots x_{L_1}$) can be aligned to the nth base y_n in RNA-2 ($\mathbf{Y} = y_1y_2\cdots y_{L_2}$), only if the following condition is satisfied:

$$\left|\frac{L_2}{L_1}m - n\right| \leq M, \qquad (1)$$

for a given M. For convenience, we refer to this constraint as the *M-constraint*. The parameter M is used to specify the maximum distance between the alignable bases. By imposing this constraint, we are restricting the number of insertions and deletions in homologous sequences, which is a reasonable assumption for real biological sequences. The constrained alignment space is band-shaped as depicted in Figure 2(a). Despite its simplicity, is has been shown that the proposed constraint works reasonably well [6, 11]. The latest implementation of Dynalign [6] takes a more principled approach for estimating the alignment region. In the new approach, a hidden Markov model (HMM) is used to estimate the set $\mathcal{Q}$ of aligned positions (m,n), whose *co-incidence probability*$P(x_m \longleftrightarrow y_n \mid \mathbf{X},\mathbf{Y})$ is larger than a reasonably low threshold $p_t(\approx 0)$:

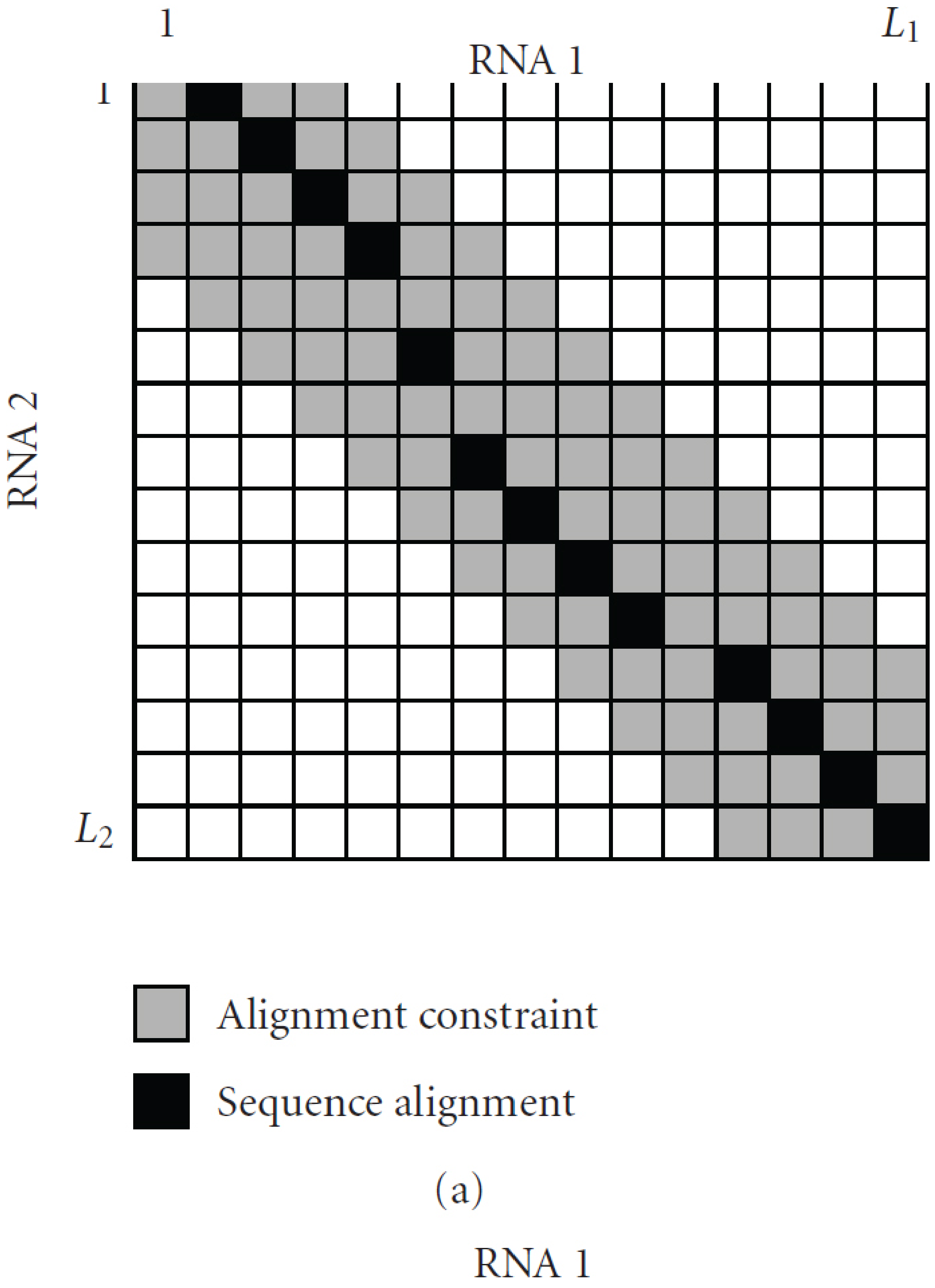
$$\mathcal{Q} = \{(m,n) \mid P(x_m y_n \mid \mathbf{X},\mathbf{Y}) > p_t\}. \qquad (2)$$

Two bases x_m and y_n are said to be coincident if (i) they are either aligned to each other, or (ii) if x_m is inserted in the region that immediately follows x_{m_1} $(m_1 < m)$ which is aligned to y_n, or vice versa [6]. The estimated set $\mathcal{Q}$ is used to constrain the final alignment. It was demonstrated that this technique can provide significant savings in computational time as well as a small improvement in alignment accuracy [6]. Another pairwise folding and alignment algorithm called *Consan* [5] first finds the confidently aligned base positions, referred to as "pins," and constrains the RNA alignment by fixing these positions. This is illustrated in Figure 2(b). The set of pins $\mathcal{P}$ is estimated using a pair-HMM, by looking for base positions (m,n) whose alignment probability $P(x_m \Leftrightarrow y_n \mid \mathbf{X},\mathbf{Y})$ exceeds some threshold p_h, which is close to unity. This set $\mathcal{P}$ can be written as follows:

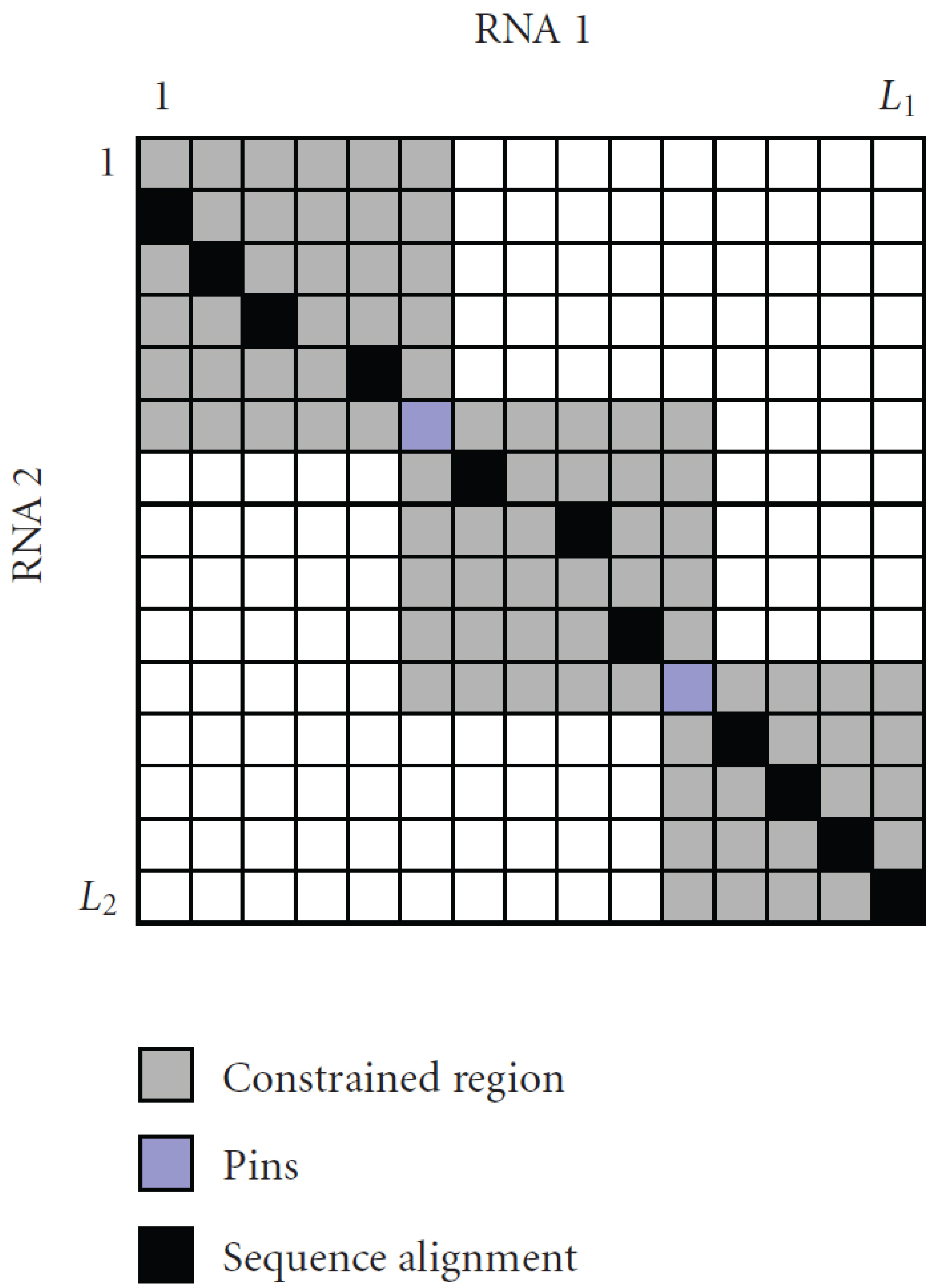
$$\mathcal{P} = \{(m,n) \mid P(x_m \Leftrightarrow y_n \mid \mathbf{X},\mathbf{Y}) \geqslant p_h\}. \qquad (3)$$

For every predicted pin $(m,n) \in \mathcal{P}$, the bases x_m and y_n are forced to be aligned to each other in the final alignment. While Dynalign [6] finds the set of *all possible* alignment positions, Consan [5] tries to find only a small set of alignment positions that *must* be included in the final alignment.

Figure 2 (a) Alignment constraint in Dynalign [11]. The maximum distance between the aligned bases is restricted. (b) Consan [5] constrains the alignment space by fixing the "pins," or confidently aligned base positions.



(a)


RNA 1
1
L_1
1
RNA 2
L_2
Constrained region
Pins
Sequence alignment

(b)

Although the previous alignment constraints [5, 6, 11] were mainly used to speed up Sankoff-style joint alignment and folding algorithms, similar ideas can be used to expedite dynamic programming alignment algorithms such as the CYK algorithm [3] for CMs and the *sequential component adjoining (SCA) algorithm* [16, 17] for profile-csHMMs. In the following section, we propose a new method for finding effective sequence alignment constraints that are especially useful for making these algorithms faster.

3. Alignment Constraints for RNA Family-Specific Models

Let us assume that we have a reference RNA whose structure is known. This can be either the consensus sequence of an RNA family or simply a single RNA sequence. Also assume that we are given a target RNA with an unknown structure, which might be a putative member of the same family. We want to find the optimal alignment between these RNAs by considering both their sequence and structural similarities. This *structural alignment* can be used for predicting the secondary structure of a new homologue [17, 20] or performing an RNA similarity search to identify new members in the same RNA family [3]. In order to find the structural alignment, we first construct a stochastic model (such as a profile-csHMM or a CM) that can closely represent the reference RNA. Then we use a dynamic programming alignment algorithm to find the best alignment between the reference RNA (represented by the constructed model) and the target RNA. Although the computational complexity of these algorithms is generally lower than that of Sankoff-style algorithms, it still ranges between $O(L^3)$ and $O(L^6)$ for a target RNA of length L. For RNAs without pseudoknots, the computational complexity of the alignment algorithm will be $O(L^3)$. The complexity for aligning pseudoknotted RNAs is at least $O(L^4)$, and it can become higher as the structure gets more complex. This renders the dynamic programming algorithms impractical for aligning long RNAs or scanning a large database, and using effective alignment constraints can be greatly helpful in relieving this problem.

3.1. Motivation for Estimating Constraints Based on Predicted Alignment Positions

When we are interested in a specific RNA family, it will be more appropriate to establish the alignment constraints based on the member sequences in the given family. Therefore, it will be more desirable to use a *family-specific* model for finding the constraints, rather than using a *general* model that applies to all RNAs as in [5, 6]. However, in many practical situations, we may not have enough number of sequences in the given family for reliably estimating the model parameters. As we can see in (2) and (3), the alignment constraints in Dynalign [6] and Consan [5] strongly depend on the estimated alignment probabilities. Although the alignment constraints used in these methods are expected to work well when we have a large number of training sequences, they are not suitable when only a handful of RNAs are available for training the model.

So, how can we find efficient alignment constraints for a family-specific model when we have only a limited number of sequences in the reference RNA family? In order to answer this question, let us consider the pair-HMMs shown in Figure 3. Both pair-HMMs have three hidden states, ALN, $\mathrm{I_X}$, and $\mathrm{I_Y}$, for base alignment, base insertion in $\mathbf{X}$ (RNA-1), and base insertion in $\mathbf{Y}$ (RNA-2), respectively. The state ALN emits a pair of aligned bases $x_m \in \mathbf{X}$ and $y_n \in \mathbf{Y}$. The insert state $\mathrm{I_X}$ emits an unaligned base x_m in $\mathbf{X}$, and similarly, the state $\mathrm{I_Y}$ emits an unaligned base y_n in $\mathbf{Y}$. These pair-HMMs can be used for finding a *sequence-based alignment* between two RNAs, and for estimating the base alignment probabilities. It is called a *sequence-based alignment*, since the alignment is obtained based on sequence similarity alone. Similar models have been used to find the alignment constraints (2) and (3) in Dynalign [6] and Consan [5], respectively. In this example, the transition probabilities of the pair-HMMs are shown along the arrows. We assume that the probability of entering a state in the beginning is identical for all three states. The emission probability $e(x,y \mid \mathrm{ALN})$ of a pair of aligned bases (x,y) is shown inside the box below the respective HMMs in Figure 3. Finally, the emission probability at an insert state is specified as follows:

$$e(x|s) = \frac{1}{4}, \quad \forall x \in \{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{U}\},\ \forall s \in \{\mathrm{I_X},\mathrm{I_Y}\}. \tag{4}$$

Now, let us assume that we want to find the sequence-based alignment of the following RNAs:

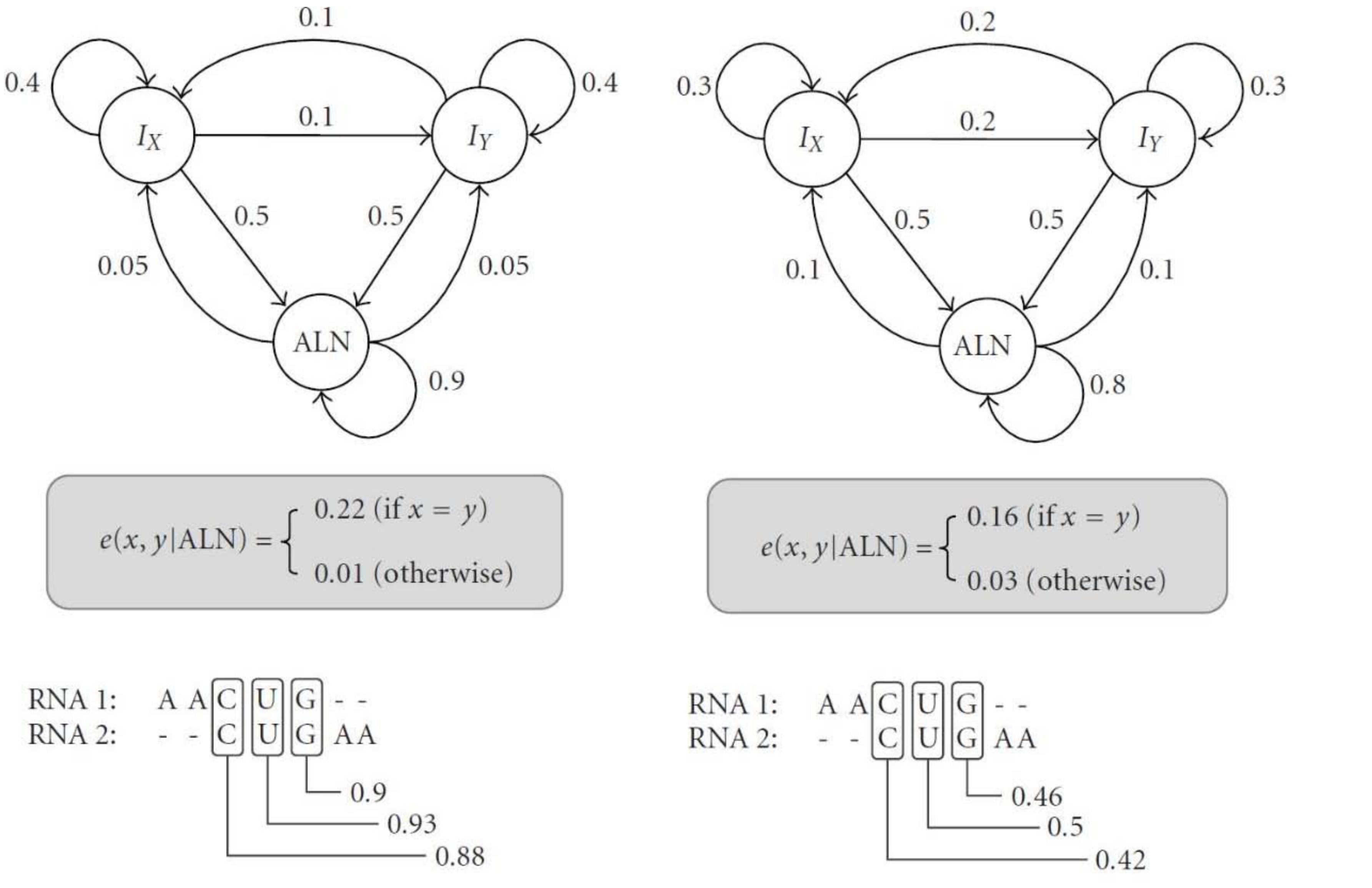
$$\begin{aligned}\mathbf{X} &= \mathrm{AACUG},\\ \mathbf{Y} &= \mathrm{CUGAA},\end{aligned} \tag{5}$$

using the pair-HMM shown in Figure 3(a). Using the Viterbi algorithm, we can get the following alignment

$$\begin{aligned}\mathbf{X} &: \mathrm{A\,A\,C\,U\,G} - -,\\ \mathbf{Y} &: - - \mathrm{C\,U\,G\,A\,A}.\end{aligned} \tag{6}$$

For each aligned pair (x_m, y_n), we can compute the alignment probability $P(x_m \Leftrightarrow y_n | \mathbf{X}, \mathbf{Y})$ using the *forward-backward algorithm* [21]. The estimated base alignment probabilities are shown in Figure 3(a), below the RNA alignment. The estimated alignment probabilities are close to unity, indicating that we can be more or less confident about the predicted base alignments. Now, let us repeat this process using the pair-HMM shown in Figure 3(b), which has slightly different parameters. As we can see in Figure 3(b), HMM-2 finds the same alignment as HMM-1, but the estimated alignment probabilities are significantly different from the previous estimates. This example clearly shows that the estimation of the base alignment probability $P(x_m \Leftrightarrow y_n | \mathbf{X}, \mathbf{Y})$ can be very sensitive to small changes in the model parameters. This implies that the alignment constraint in (3), which depends on $P(x_m \Leftrightarrow y_n | \mathbf{X}, \mathbf{Y})$, may not be reliable when we do not have enough training data to accurately estimate the HMM parameters. Compared to this, the alignment constraint in (2) might be more reliable, as the coincidence probability $P(x_m \longleftrightarrow y_n | \mathbf{X}, \mathbf{Y})$ used to estimate the constraint also includes the insertion probabilities. However, the predicted constraint will nevertheless depend on the parameters of the HMM to a considerable extent.

Figure 3 Two pair-HMMs with slightly different parameters. Both pair-HMMs have three states, ALN, I_X, and I_Y, which represent base alignment, base insertion in RNA-1, and base insertion in RNA-2, respectively.HMM-1HMM-2


0.1
0.4
0.4
0.1
I_X
I_Y
0.5
0.5
0.05
0.05
ALN
0.9
$e(x, y|\text{ALN}) = \begin{cases} 0.22 \text{ (if } x = y) \\ 0.01 \text{ (otherwise)} \end{cases}$
RNA 1: A A C U G - -
RNA 2: - - C U G A A
0.9
0.93
0.88
0.2
0.3
0.3
0.2
I_X
I_Y
0.5
0.5
0.1
0.1
ALN
0.8
$e(x, y|\text{ALN}) = \begin{cases} 0.16 \text{ (if } x = y) \\ 0.03 \text{ (otherwise)} \end{cases}$
RNA 1: A A C U G - -
RNA 2: - - C U G A A
0.46
0.5
0.42

(a) HMM-1

(b) HMM-2

However, one thing we can notice by comparing Figures 3(a) and 3(b) is that, despite the large difference in the estimated alignment probabilities, the resulting sequence alignments are identical. In fact, the alignment positions in an optimal sequence alignment are not very sensitive to small parameter changes, and as a result, HMMs with reasonably similar parameters often yield almost identical alignment results. This motivates us to exploit the *aligned base positions*(x_m, y_n) for establishing the alignment constraints, instead of using the *base alignment probabilities*$P(x_m \Leftrightarrow y_n \mid \mathbf{X}, \mathbf{Y})$.

3.2. Finding Seed Regions Using a Profile-HMM

Based on the previous observation, we propose a new method that utilizes the predicted alignment positions to find the alignment constraints. As before, let us denote the structured reference RNA as $\mathbf{X}$ (RNA-1) and the unstructured target RNA as $\mathbf{Y}$ (RNA-2). Ultimately, we want to find the *structural alignment* of these RNAs. However, since the dynamic programming algorithm for finding the structural alignment is computationally expensive, we want to come up with effective alignment constraints that can speed up the alignment.

For this purpose, we first build a profile-HMM [3] based on the reference RNA family. This model is used to find the *sequence-based alignment* between the reference RNA (represented by the profile-HMM) and the target RNA. Secondly, we identify the regions that consist of multiple consecutive base alignments, or base matches. Although a single base match may not be meaningful by itself, having a region of consecutive matches often indicates that the alignment in the given region is reasonably accurate. This is especially true for those matches that are located in the middle of the region. For example, we can see in both Figures 3(a) and 3(b) that the alignment probability $P(x_m \Leftrightarrow y_n \mid \mathbf{X}, \mathbf{Y})$ is the largest for the alignment between $x_4 = \mathrm{U}$ and $y_2 = \mathrm{U}$, which is located between the matches (x_3, y_1) and (x_5, y_3). Therefore, we exclude the matches near the end and keep the remainder to obtain a set of reliable base alignments. The set of reliable contiguous matches is referred to as the *seed region*. The procedure for finding the seed regions can be summarized as follows.

(1) Find a sequence-based alignment between the RNAs.

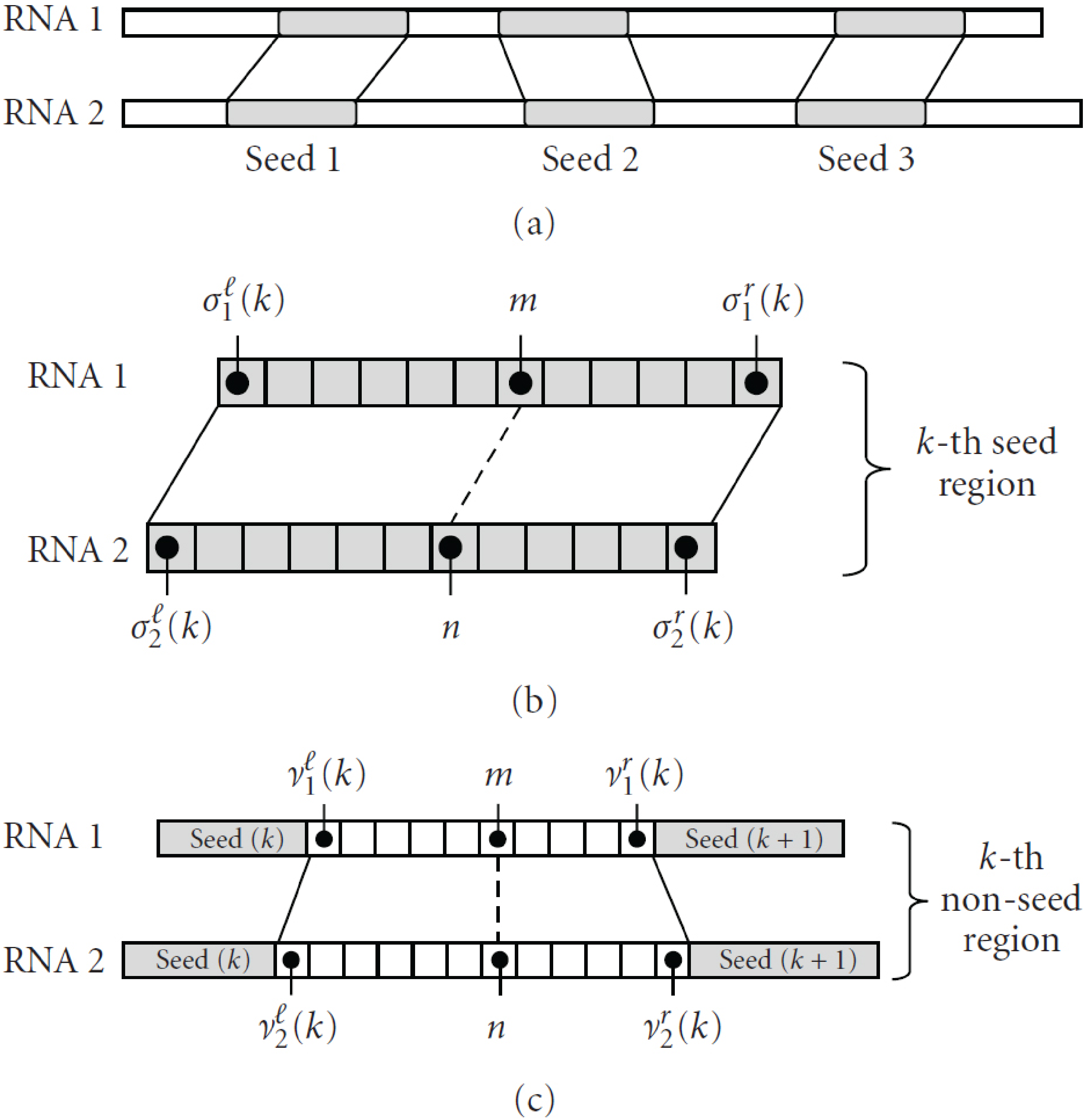
(2) Identify all regions, and longest such regions, that consist of consecutive matches. Let N_{match} be the number of consecutive matches in a given region. Keep only those regions with $N_{\mathrm{match}} \geq \Gamma$.

(3) In each region, exclude the first γ matches in the left end and the last γ matches in the right end.

(4) The region that consists of the $N_{\mathrm{match}} - 2\gamma$ remaining matches is defined as a seed region.

The integer parameters Γ and $\gamma\,(< \Gamma/2)$ define the seed regions during this process. In general, using a larger Γ will identify a smaller number of seed regions, and a larger γ makes the seed regions contain fewer but more reliable base matches. Figure 4(a) illustrates an example alignment with three seed regions.

Figure 4 Illustration of the proposed method. (a) Seed regions are identified from the sequence-based alignment. (b) Example of a seed region, which consists of consecutive base matches. (c) Example of a nonseed region.



3.3. Constraints in a Seed Region

Assume that we have identified K seed regions according to the procedure described in Section 3.2. Since the base alignments in these regions are relatively reliable, we keep the alignment space in these regions small. Let us consider the kth seed region illustrated in Figure 4(b). We denote the begin index and the end index of the kth seed in RNA-1 as $\sigma_1^{\ell}(k)$ and $\sigma_1^{r}(k)$, respectively. The superscripts ℓ and r in $\sigma_1^{\ell}(k)$ and $\sigma_1^{r}(k)$ stand for "left" and "right," respectively. Similarly, the begin and the end indices of the kth seed in RNA-2 are denoted as $\sigma_2^{\ell}(k)$ and $\sigma_2^{r}(k)$, respectively. Since a seed region consists of consecutive base matches, we have

$$\sigma_1^{r}(k) - \sigma_1^{\ell}(k) + 1 = \sigma_2^{r}(k) - \sigma_2^{\ell}(k) + 1 = L_k, \tag{7}$$

where L_k is the length of the kth seed. For convenience, we define D_k as the position difference between the aligned bases in the kth seed

$$\sigma_2^{\ell}(k) - \sigma_1^{\ell}(k) = \sigma_2^{r}(k) - \sigma_1^{r}(k) = D_k. \tag{8}$$

Based on the kth seed, we define the set of allowed alignment positions (m,n) as follows:

$$\mathcal{S}(k) = \{(m,n) \mid \sigma_1^{\ell}(k) \leq m \leq \sigma_1^{r}(k), |n - m - D_k| \leq \Delta\}. \tag{9}$$

For a base x_m that is contained in the kth seed region of RNA-1, the parameter Δ restricts the distance between the base y_{n^*} ($n^* = m + D_k$), to which x_m is aligned in the sequence-based alignment, and the base y_n, to which x_m will be aligned in the final structural alignment. As the base alignments in the seed regions are reliable, Δ can be typically set to a small number. We find the set $\mathcal{S}(k)$ for all $k = 1, 2, \ldots, K$, and these sets will be combined later to establish the final alignment constraints.

3.4. Constraints in a Nonseed Region

The predicted base alignments in the nonseed regions are generally less reliable compared to those inside the seed regions. Therefore, we define different alignment constraints for the bases contained in the nonseed regions, and make the constraints less stringent compared to (9). Let us consider the kth nonseed region illustrated in Figure 4(c). The begin and end indices of the kth nonseed region in RNA-1 are denoted by $\nu_1^{\ell}(k)$ and $\nu_1^{r}(k)$, respectively. Similarly, $\nu_2^{\ell}(k)$ and $\nu_2^{r}(k)$, respectively denote the begin and end indices of the corresponding nonseed region in RNA-2. Now, we define the set $\mathcal{A}(k)$, which contains (i) all aligned base positions (m,n) in the kth nonseed region, as well as (ii) the first and last positions $(\nu_1^{\ell}(k), \nu_2^{\ell}(k))$ and $(\nu_1^{r}(k), \nu_2^{r}(k))$ in this region:

$$\begin{aligned} \mathcal{A}(k) = \{(m,n) | \nu_1^{\ell}(k) \leq m \leq \nu_1^{r}(k), \nu_2^{\ell}(k) \\ \leq n \leq \nu_2^{r}(k), x_m \Leftrightarrow y_n\} \\ \cup \{(\nu_1^{\ell}(k), \nu_2^{\ell}(k)), (\nu_1^{r}(k), \nu_2^{r}(k))\}. \end{aligned} \tag{10}$$

In practice, it is possible that there may be no aligned bases $x_m \Leftrightarrow y_n$ in the given nonseed region. Including the terminal positions $(\nu_1^{\ell}(k), \nu_2^{\ell}(k))$ and $(\nu_1^{r}(k), \nu_2^{r}(k))$ of the kth nonseed region in $\mathcal{A}(k)$ ensures that the set $\mathcal{A}(k)$ will never be empty. For the position pairs $(m,n) \in \mathcal{A}(k)$, we estimate the range of the position difference $(n - m)$ as follows:

$$\begin{aligned} \Delta_{\min}(k) &= \min_{(m,n) \in \mathcal{A}(k)} (n - m), \\ \Delta_{\max}(k) &= \max_{(m,n) \in \mathcal{A}(k)} (n - m). \end{aligned} \tag{11}$$

Based on these values, we define the following set:

$$\begin{aligned} \mathcal{N}(k) = \{(m,n) | \nu_1^{\ell}(k) \leq m \leq \nu_1^{r}(k), m + \Delta_{\min}(k) - \Delta \\ \leq n \leq m + \Delta_{\max}(k) + \Delta\}, \end{aligned} \tag{12}$$

which contains the alignable base positions (m,n) in the kth nonseed region. Note that we use the same Δ in (9) and (12). Therefore, a larger Δ will relax the alignment constraints for both seed and nonseed regions, and a smaller Δ will make both constraints more stringent.

3.5. Overall Alignment Constraints

In Sections 3.3 and 3.4, we defined the alignment constraints in the seed regions as well as the constraints in the nonseed regions. Finally, we combine (9) and (12) to obtain the overall alignment constraints $\mathcal{C}$ as follows:

$$= \left[\bigcup_{\forall k_1} (k_1)\right] \cup \left[\bigcup_{\forall k_2} (k_2)\right]. \qquad (13)$$

This set $\mathcal{C}$ can be used to constrain the alignment space of the dynamic programming algorithm for finding the structural alignment of the given RNAs. When finding the RNA alignment, we allow a base x_m in RNA-1 to be aligned to a base y_n in RNA-2 only if the pair (m,n) is included in this set $\mathcal{C}$.

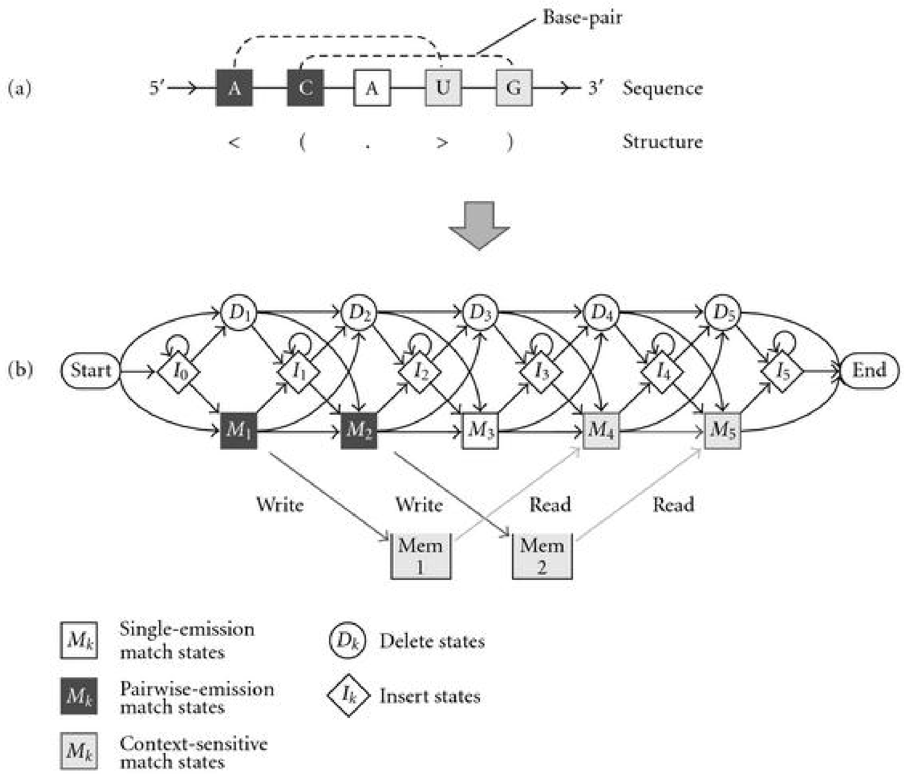
4. Experimental Results

To demonstrate the effectiveness of the proposed method, we applied the new alignment constraints to the profile-csHMM-based structural alignment method [17]. In the following, we provide a brief explanation about the experimental set-up and present the experimental results.

4.1. Profile-csHMM-Based Structural Alignment

Profile-csHMMs are a subclass of context-sensitive HMMs [22] that are especially useful for representing RNA sequence profiles and their secondary structure. In principle, profile-csHMMs can represent RNA secondary structures with any kind of base pairs [16, 17]. As a result, profile-csHMMs can also be used for aligning and predicting the structure of RNAs that contain pseudoknots, which cannot be done using the widely used SCFGs (or CMs). The profile-csHMM-based structural alignment algorithm proposed in [17] proceeds as follows. In the first place, a profile-csHMM is constructed based on a reference RNA sequence with a known structure. In [17], a single reference RNA was used to build the model. This can be used for performing a single RNA homology search, similar to the CM-based search proposed in [23]. Figure 5 illustrates an example, where a profile-csHMM is constructed based on a reference RNA that has two crossing base pairs. Obviously, we do not have enough training sequences to accurately estimate the model parameters in this case, hence the parameters of the profile-csHMM are chosen according to the scoring scheme proposed by Gorodkin et al. [24]. These scores can be viewed as normalized log-probabilities for observing base substitutions or gaps (insertions and deletions) in homologous RNAs. They have been used in a number of RNA alignment algorithms [20, 24], yielding accurate alignment results. The constructed profile-csHMM can then be used for finding the optimal structural alignment between the reference RNA and an unstructured target RNA, computing their alignment score, and predicting the secondary structure of the target RNA. A dynamic programming alignment algorithm called the *sequential component adjoining (SCA) algorithm* can be used for this purpose.

Figure 5 Constructing a profile-csHMM. (a) A reference RNA sequence with a known secondary structure. (b) The profile-csHMM that represents the given reference RNA.



4.2. Estimating the Alignment Constraints

In order to estimate the alignment constraints for expediting the profile-csHMM alignment algorithm (or the SCA algorithm), we construct a profile-HMM-based on the same reference RNA. Note that unlike the profile-csHMM, the traditional profile-HMM reflects only the sequence characteristics of the reference RNA. Similar to the parameterization of the profile-csHMM described in Section 4.1, the parameters of the profile-HMM are also specified according to the scores in [24]. The resulting profile-HMM is used to estimate the sequence alignment constraints as we elaborated in Section 3. We use the estimated constraints to restrict the alignment space of the structural RNA alignment to reduce the overall computational load.

4.3. Choosing the Parameters for Constraint Estimation

Now, one practical question is how we should choose the parameters Γ, γ, and Δ that are used to estimate the alignment constraints in Sections 3.3 and 3.4. Ideally, the predicted alignment constraints should minimize the alignment space without affecting the quality of the final structural alignment. Since the alignment constraints proposed in Section 3 are derived from the predicted seed regions, the alignment accuracy in these regions has a crucial impact on the accuracy of the proposed approach. For this reason, we estimated the average base alignment accuracy in the seed regions for the 5S rRNA and tRNA families in the Rfam database (version 8.1) [25]. We used the RNAs in the *seed alignment* of the respective family, as they have a relatively reliable secondary structure annotation. For each RNA family, we first chose a reference RNA among its members, and constructed a profile-HMM based on the chosen RNA. Then we aligned the remaining members to the reference RNA using the profile-HMM. For every sequence alignment, the predicted alignment positions have been compared to the correct positions in the database to estimate the alignment error rate. In order to get a reliable estimate, we repeated these experiments by using every member as the reference RNA. This resulted in 1, 182, 656 pairwise alignments for tRNAs and 345, 156 alignments for 5S rRNAs.

Let us assume that the profile-HMM predicted that x_m in the reference RNA should be aligned to y_n in the target RNA. We want to estimate the probability of error for this prediction as a function of the following parameters:

(1) : the number of consecutive base matches in the region containing the alignment (m,n),

(2) : the minimum distance between the given base alignment (m,n) and the terminal alignment positions (m_1,n_1) and (m_2,n_2). See Figure 6 for illustration.

Consider the example illustrated in Figure 6. The number of consecutive base matches, or the *length*, of the region containing (m,n) is

$$L_{\mathrm{aln}} = m_2 - m_1 + 1 = n_2 - n_1 + 1, \tag{14}$$

and the minimum distance d_{min} is defined as

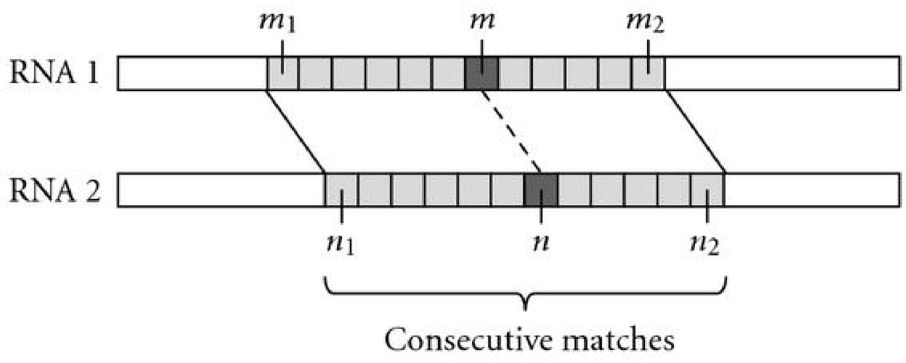
$$\begin{aligned} d_{\mathrm{min}} &= \min\left((m - m_1), (m_2 - m)\right) \\ &= \min\left((n - n_1), (n_2 - n)\right). \end{aligned} \tag{15}$$

Based on these definitions, we define

$$\begin{aligned} &P_e(L,d) \\ &= P\left(x_m \not\Leftrightarrow y_n \text{ in the trusted alignment} \mid L_{\mathrm{aln}} \geq L, d_{\mathrm{min}} \geq d\right). \end{aligned} \tag{16}$$

This is the probability that the predicted base alignment (m,n) between x_m and y_n will be incorrect, given that the following hold:

Figure 6 A region that consists of consecutive base matches.

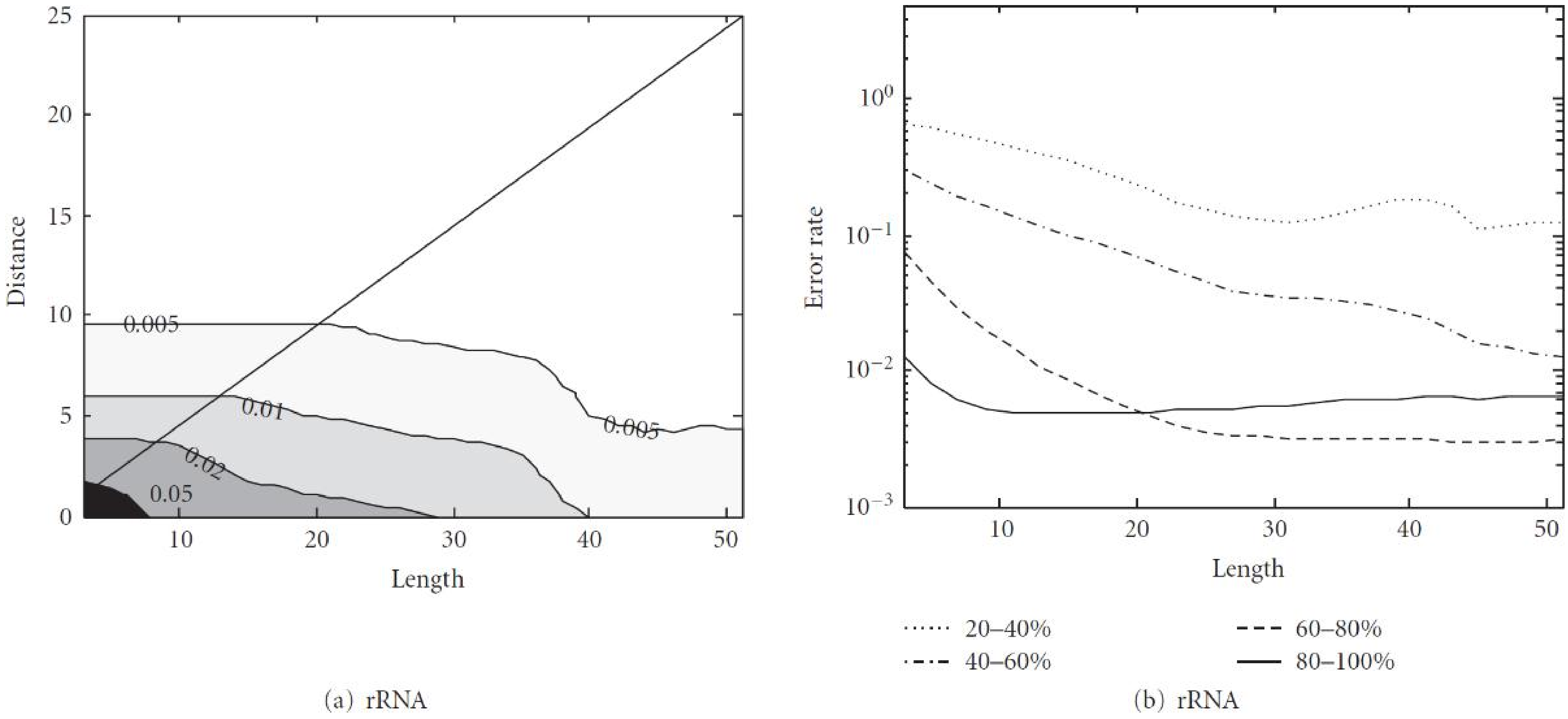


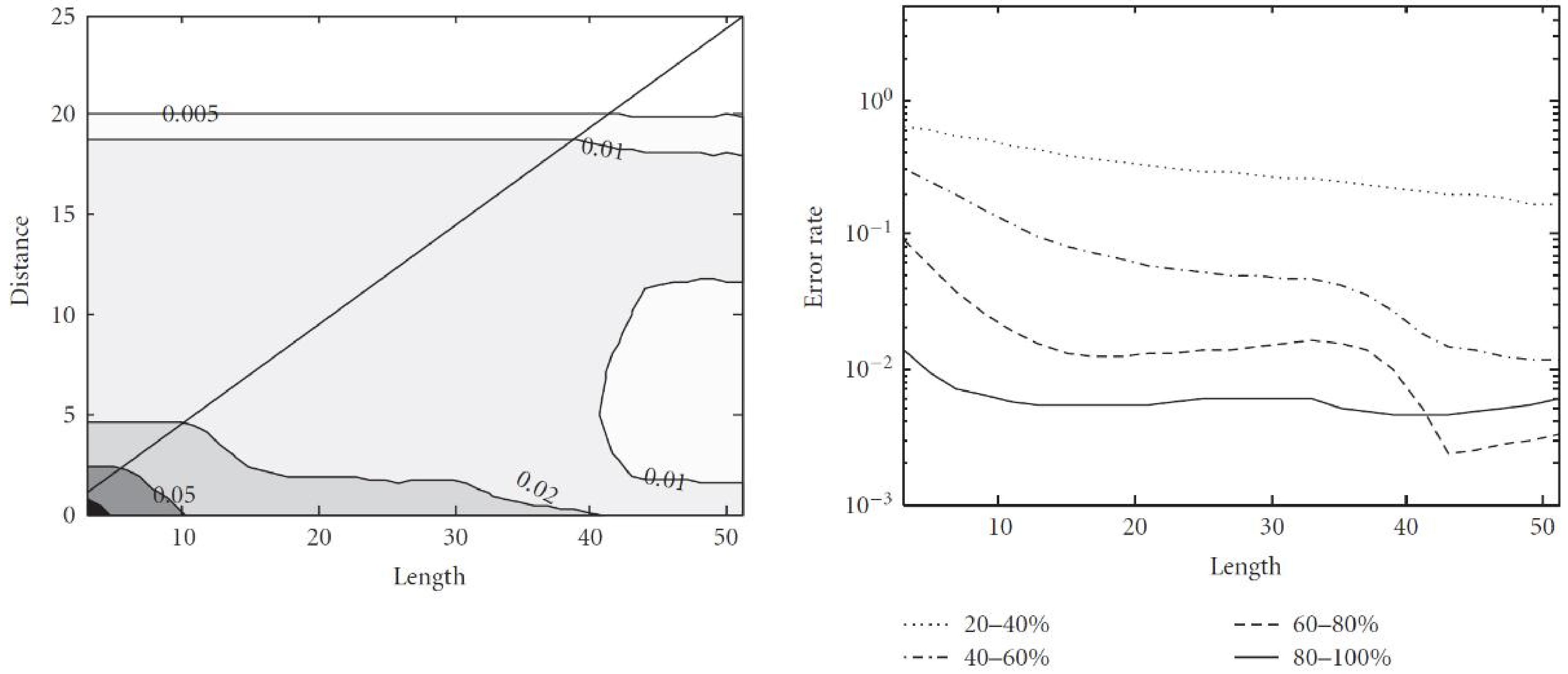
(1) the length L_{aln} of the alignment region containing (m,n) is at least L,

(2) there are at least d matches in the left-hand side of (m,n) as well as in the right-hand side.

Figure 7(a) shows the contour plot of the misalignment probability $P_e(L,d)$ for 5S rRNAs, where the x-axis is for L and the y-axis is for d. On top of each contour curve, we show the corresponding misalignment probability $P_e(L,d)$ for the points (L,d) on the given curve. Darker shaded regions correspond to higher $P_e(L,d)$ and lighter shaded regions correspond to lower $P_e(L,d)$. The diagonal line representing $L = 2d + 1$ is shown in the plot as a reference. Note that, by definition, we have $L \geq 2d + 1$. Therefore, for any (L,d) such that $L < 2d + 1$, which corresponds to the region above the diagonal line, we will have $P_e(L,d) = P_e(L,(L-1)/2)$.

Figure 7 Alignment error probability $P_e(L,d)$ in the seed regions. (a) Contour plot of $P_e(L,d)$ for 5S rRNAs. Darker shade corresponds to higher $P_e(L,d)$ and lighter shade corresponds to lower $P_e(L,d)$. The misalignment probability $P_e(L,d)$ on the level curves are shown on top of the contours. (b) Misalignment probability for 5S rRNAs with different percentage identities. (c) Contour plot of $P_e(L,d)$ for tRNAs. (d) Misalignment probability for tRNAs with different percentage identities.rRNArRNAtRNAtRNA





(c) tRNA

(d) tRNA

As we would expect, the misalignment probability $P_e(L,d)$ becomes smaller as L and d get larger. Figure 7(b) shows the misalignment probability $P_e(L,d)$ of 5S rRNAs for $L = 2d + 1$. The pairwise alignments have been divided into different groups based on the *percentage identity* (or *percent sequence similarity*) of the aligned RNAs, and the alignment error probability $P_e(L,d)$ has been computed for the respective groups. As we can see in Figure 7(b), the error probability is generally lower for RNAs with higher percentage identity. This is expected, since the seed regions are predicted from a sequence-based alignment, which will be more accurate if the RNAs have higher sequence similarity. Figures 7(c) and 7(d) show the misalignment probability $P_e(L,d)$ for tRNAs, which have similar trends.

We also computed $P_e(L,d)$ for RNAs with $60\% \sim 100\%$ identity for different values of (L,d). This is summarized in Table 1. For example, the alignment error probability $P_e(L,d)$ for 5S rRNAs is 1.81% for $(L,d) = (9,4)$ and 0.79% for $(L,d) = (15,7)$. This implies that if we choose $\Gamma = 9$ and $\gamma = 4$ when finding the seed regions (see Section 3.2), more than 98% of the alignments in the predicted seed regions will be correct. In our experiments, we observed that the misaligned bases were typically located within $1 \sim 2$ base positions from the correct ones. This implies that if we let $\Delta = 1$ or $\Delta = 2$, most of the correct base alignments (m,n) will be included in the constrained alignment space $\mathcal{S}(k)$ in (9). Therefore, imposing these constraints will not degrade the accuracy of the final structural alignment.

Table 1 Misalignment probability in the seed regions of RNAs with 60% $\sim$ 100% identity. The probability $P_e(L,d)$ has been estimated for five different values of (L,d).

	(L,d)				
	(7,3)	(9, 4)	(11, 5)	(13, 6)	(15, 7)
	Error (%)	Error (%)	Error (%)	Error (%)	Error (%)
5S ribosomal RNA	2.64	1.81	1.32	1.00	0.79
Transfer RNA	3.60	2.46	1.80	1.44	1.28

4.4. RNA Structural Alignment with the Proposed Constraints

As mentioned earlier, we applied the proposed constraints to the profile-csHMM-based structural alignment algorithm [17]. For our experiments, we chose six RNA families from the Rfam database [25]. The Flavi_pk3 RNAs were obtained from Rfam 7.0, which are now part of the Flavi_CRE family in Rfam 8.1. For other families, we obtained the RNAs from Rfam 8.1. Among the six families, two families, tRNAs and 5S rRNAs, do not contain pseudoknots in their secondary structures, while the other four families, Corona_pk3, HDV_ribozyme, Tombus_3IV, and Flavi_pk3, contain pseudoknots. The basic properties of these RNA families, such as the number of RNAs in the Rfam seed alignment, the average length of the member sequences, and their average percentage (sequence) identity, are shown in Table 2. For each RNA family, we performed the following experiment.

Table 2 Basic properties of the RNA families used in the experiments

	Number of seed sequences	Average length	Average percentage identity
Transfer RNA	1088	72.7	45
5S ribosomal RNA	602	116.8	61
Corona_pk3	14	62.5	70
HDV_ribozyme	15	88.8	95
Tombus_3_IV	18	64.5	94
Flavi_pk3	14	95.4	69

(1) Choose a reference RNA from the seed alignment.

(2) Construct a profile-HMM and a profile-csHMM-based on the reference RNA.

(3) Choose a different target RNA from the seed alignment.

(4) Estimate the alignment constraint using the profile-HMM.

(5) Apply the alignment constraint and find the structural alignment using the profile-csHMM.

(6) Repeat step-3 to step-5 for different target RNAs.

(7) Repeat step-1 to step-6 for different reference RNAs.

In order to measure the quality of the structural alignment, we predicted the secondary structure of the target RNA based on the structural alignment, and compared it to the trusted structure in the Rfam database. Then we counted the number of correctly predicted base pairs (TP; true-positives), the number of incorrectly predicted base pairs (FP; false-positives), and the number of true base pairs that could not be predicted (FN; false-negatives). Based on these numbers, we estimated the *sensitivity (SN)* and the *positive predictive value (PPV)* as follows:

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \qquad \mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{17}$$

The sensitivity is defined as the fraction of base pairs in the trusted structure that could be predicted by the algorithm, and the positive predictive value is defined as the fraction of predicted base pairs that were correct.

We first tested the performance of the proposed approach on RNA families that do not contain pseudoknots. In order to compare the effectiveness of different alignment constraints, we repeated the above experiment for the following methods.

(1) Profile-csHMM + proposed alignment constraint (referred to as "PROPOSED").

(2) Profile-csHMM + *M*-constraint (referred to as "*M*-CONSTRAINT").

(3) Profile-csHMM (original implementation in [17]; referred to as "ORIGINAL").

Table 3 summarizes the average sensitivity (SN), positive predictive value (PPV), and alignment time for using different alignment constraints with the profile-csHMM-based structural alignment method. The CPU time for finding the alignment has been measured on a MacPro with two 2.8 GHz quad-core Intel Xeon processors and 4 GB memory. These results have been obtained from one thousand structural alignments of distinct pairs of RNAs that were chosen from the seed alignment of the respective RNA families. For these experiments, we used $(\Gamma,\gamma,\Delta) = (9,4,2)$ for tRNAs and $(\Gamma,\gamma,\Delta) = (9,4,0)$ for 5S rRNAs. These parameters were chosen based on the analysis in Section 4.3. In general, there will be a tradeoff between alignment accuracy and runtime. These parameters have been used as they provide a good balance between these two measures. Further discussion on this tradeoff can be found at the end of Section 4.4. For the *M*-constraint defined in (1), we used $M = 7$ as in [6]. As we can see in Table 3, all three methods were able to achieve accurate alignment results that were comparable to each other. However, adopting the proposed alignment constraint improved the average alignment speed significantly, which was around 7 $\sim$ 68 times faster compared to the fixed *M*-constraint, and up to 2.4 times faster compared to the original implementation in [17] that uses a simple heuristic.

Table 3 Average sensitivity (SN), positive predictive value (PPV), and alignment time for RNA families that do not contain pseudoknots

	Profile-csHMM								
	***M* -constraint**			**Original**			**Proposed**		
	SN (%)	**PPV (%)**	**Time (sec)**	**SN (%)**	**PPV (%)**	**Time (sec)**	**SN (%)**	**PPV (%)**	**Time (sec)**
Transfer RNA	94.2	95.8	0.0739	94.1	96.0	0.0139	93.6	96.2	0.0108
5S ribosomal RNA	94.8	96.3	0.0676	95.1	97.0	0.0024	95.9	98.5	0.0010

In order to test the performance of the proposed method on RNA pseudoknots, we carried out similar experiments using four pseudoknotted RNA families, Corona_pk3, HDV_ribozyme, Tombus_3_IV and Flavi_pk3. For these experiments, we used $(\Gamma,\gamma,\Delta) = (9,4,0)$ and $M = 3$ for all four RNA families. In addition to evaluating the performance of the profile-csHMM method for these families, we evaluated the performance of the PSTAG-based method [20] for comparison. The PSTAG-based structural alignment method is a state-of-the-art pairwise RNA alignment method that uses *pair stochastic tree adjoining grammars (PSTAGs)*. PSTAGs can be used for aligning many known pseudoknots, though not all of them. To the best of our knowledge, the PSTAG-based alignment method [20] is the only grammar-based method that can be used for finding the structural alignment of pseudoknotted RNAs, except for the profile-csHMM method. Table 4 shows the average sensitivity and positive predictive value of the different alignment methods. The sensitivity and the positive predictive value of the PSTAG-based method have been obtained from [20] based on the same test set. As we can see in this table, all four methods could achieve high sensitivity and PPV for the Corona_pk3, HDV_ribozyme, and Tombus_3_IV RNA families. The Flavi_pk3 RNAs could not be aligned using PSTAGs, as they have a more complex secondary structure compared to other RNA families. Unlike PSTAGs, profile-csHMMs can handle RNAs with any kind of base pairs, hence they could be used for aligning Flavi_pk3 RNAs as well. The current implementation can handle any RNAs in the Rivas and Eddy class [26], which includes nearly all known pseudoknots. We can also handle the RNAs outside the Rivas and Eddy class by incorporating additional *adjoining rules*. See [19] for further discussions on adjoining rules and the descriptive capability of profile-csHMMs. Table 4 shows that all three profile-csHMM-based approaches yielded accurate alignment results for Flavi_pk3 RNAs. By comparing the performance of the profile-csHMM method with different constraints, we can note that incorporating the proposed alignment constraint virtually did not affect the alignment accuracy.

Table 4 Average sensitivity (SN) and positive predictive value (PPV) for RNA families with pseudoknots

	Profile-csHMM							
	M **-constraint**		**Original**		**Proposed**		**PSTAG**	
	SN (%)	**PPV (%)**	**SN (%)**	**PPV (%)**	**SN (%)**	**PPV (%)**	**SN (%)**	**PPV (%)**
Corona_pk3	95.5	95.7	95.7	96.5	94.8	96.0	94.6	95.5
HDV_ribozyme	94.5	95.1	94.5	95.3	94.2	95.9	94.1	95.6
Tombus_3_IV	95.9	96.4	95.9	96.4	96.8	97.4	97.4	97.4
Flavi_pk3	94.6	96.5	94.5	96.4	94.5	96.8	N/A	N/A

As we can see in Table 5, the proposed sequence alignment constraint was able to significantly improve the alignment speed also for pseudoknotted RNAs. In fact, by comparing the results in Tables 3 and 5, we can observe that the overall computational gain becomes even larger for RNAs with more complicated secondary structures. The new constraint made the alignment speed around 40 $\sim$ 100 times faster compared to the fixed *M*-constraint (using $M = 3$), and around 3 $\sim$ 11 times faster compared to the original implementation [17], at a comparable prediction accuracy. We can also note that the PSTAG-based alignment takes considerably longer than the profile-csHMM-based alignment. The large difference in alignment speed is mainly due to the fact that the PSTAG algorithm [20] does not incorporate any constraint to restrict the alignment space.

Table 5 Average CPU time for finding the structural alignment of RNAs containing pseudoknots

	Profile-csHMM			PSTAG
	M **-constraint**	**Original**	**Proposed**	
	Time (sec)	**Time (sec)**	**Time (sec)**	**Time (sec)**
Corona_pk3	9.37	0.71	0.23	19.65
HDV_ribozyme	10.30	1.03	0.13	158.77
Tombus_3_IV	6.99	0.35	0.07	193.06
Flavi_pk3	13.31	3.96	0.35	N/A

It would be also interesting to see how the parameters used for predicting the alignment constraint would affect the overall performance. For this purpose, we repeated the previous experiment using different values of Γ and γ. Three pairs of (Γ,γ) were chosen based on the experimental results shown in Figure 7, such that the average misalignment probability does not exceed 5% for both tRNAs and 5S rRNAs. We used $\Gamma = 2\gamma + 1$ in all three cases, such that the minimum length of the seed region is one. Note that if there are regions with more than Γ consecutive matches in the sequence-based alignment, the lengths of the corresponding seed regions will be longer than this minimum. In all three experiments, the parameter Δ was set to zero. Table 6 shows the sensitivity, PPV, and alignment time for different pairs of (Γ,γ). In general, small Γ and γ tend to increase the fraction of bases included in the seed regions, thereby reducing the overall alignment space. As a consequence, the alignment time becomes smaller as we can see in Table 6. However, if these values are made too small, the resulting alignment space can become too restricted, hence degrading the alignment accuracy. This phenomenon could be observed when aligning the Corona_pk3 RNAs with $\Gamma = 7$ and $\gamma = 3$.

Table 6 Performance of the proposed approach for different parameters

	Profile-csHMM (Proposed)								
	SN (%)	**PPV (%)**	**Time (sec)**	**SN (%)**	**PPV (%)**	**Time (sec)**	**SN (%)**	**PPV (%)**	**Time (sec)**
Corona_pk3	92.9	94.4	0.139	94.8	96.0	0.232	95.3	96.2	0.278
HDV_ribozyme	93.2	95.7	0.131	94.2	95.9	0.133	94.5	95.5	0.147
Tombus_3_IV	96.5	97.1	0.065	96.8	97.4	0.068	96.6	97.3	0.069
Flavi_pk3	94.8	97.2	0.329	94.5	96.8	0.351	94.5	96.8	0.362

In Tables 3 and 5, we have shown that the proposed alignment constraint can significantly reduce the average computational requirement for finding the RNA structural alignments. Since the proposed method estimates the constraint based on the sequence alignment of the given RNAs, the actual reduction in complexity will depend on the degree of sequence similarity between the RNAs. Suppose we have a reference RNA of length L_r and a target RNA of length L_t. In the best case, when these RNAs are perfectly aligned, the overall computational cost will be dominated by the constraint estimation step, hence the resulting complexity will be $O(L_r L_t)$. In the worst case, the complexity will be identical to that of an unconstrained profile-csHMM alignment, which is $O(L_r L_t^3)$ for RNAs without pseudoknots, $O(L_r L_t^4)$ for typical RNA pseudoknots (including Corona_pk3, HDV_ribozyme, Tombus_3_IV, and Flavi_pk3 used in our experiments), and $O(L_r L_t^6)$ for RNAs with the most complicated secondary structure in the Rivas and Eddy work [26]. In general, the maximum distance between the alignable bases will be limited by the constraint (12). If we define D as

$$D = 2\Delta + \max_k (\Delta_{\max}(k) - \Delta_{\min}(k)), \tag{18}$$

the computational complexity of the profile-csHMM alignment method with the proposed constraint will be $O(L_r L_t + L_r D^3)$ for RNAs that do not contain pseudoknots. For pseudoknotted RNAs in the Rivas and Eddy work, the complexity will range between $O(L_r L_t + L_r D^4)$ and $O(L_r L_t + L_r D^6)$.

5. Concluding Remarks

In this paper, we proposed a new method for finding an effective alignment constraint for fast and accurate structural alignment of RNAs. The proposed method is especially useful for accelerating the dynamic programming alignment algorithm of family-specific models, such as the profile-csHMMs or CMs. The alignment constraint proposed in this paper is not very sensitive to small parameter changes in the model that is used to predict the constraint. Therefore, it can be especially useful when we do not have enough sequences in the reference RNA family for training the model. We applied the new constraint to the profile-csHMM-based structural alignment method [17], and evaluated its performance using several RNA families containing pseudoknots. Experimental results showed that the proposed alignment constraint could significantly reduce the alignment time without any loss of alignment accuracy. Although we have mainly focused on incorporating the proposed constraint into the profile-csHMM-based method, these constraints can certainly be used to expedite other alignment methods based on CMs [3, 23] or PSTAGs [20].

References

1. Eddy SR: **Non-coding RNA genes and the modern RNA world.** *Nature Reviews Genetics* 2001,**2**(12):919-929. 10.1038/35103511

2. Yoon B-J, Vaidyanathan PP: **Computational identification and analysis of noncoding RNAs—unearthing the buried treasures in the genome.** *IEEE Signal Processing Magazine* 2007,**24**(1):64-74.

3. Durbin R, Eddy S, Krogh A, Mitchison G: *Biological Sequence Analysis*. Cambridge University Press, Cambridge, UK; 1998.

4. Sankoff D: **Simultaneous solution of the RNA folding, alignment, and protosequence problems.** *SIAM Journal on Applied Mathematics* 1985,**45**(5):810-825. 10.1137/0145048

5. Dowell RD, Eddy SR: **Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints.** *BMC Bioinformatics* 2006., **7, article 400:**

6. Harmanci AO, Sharma G, Mathews DH: **Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign.** *BMC Bioinformatics* 2007., **8, article 130:**

7. Havgaard JH, Lyngsø RB, Stormo GD, Gorodkin J: **Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%.** *Bioinformatics* 2005,**21**(9):1815-1824. 10.1093/bioinformatics/bti279

8. Havgaard JH, Torarinsson E, Gorodkin J: **Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix.** *PLoS Computational Biology* 2007,**3**(10):e193. 10.1371/journal.pcbi.0030193

9. Holmes I: **Accelerated probabilistic inference of RNA structure evolution.** *BMC Bioinformatics* 2005., **6, article 73:**

10. Mathews DH: **Predicting a set of minimal free energy RNA secondary structures common to two sequences.** *Bioinformatics* 2005,**21**(10):2246-2253. 10.1093/bioinformatics/bti349

11. Uzilov AV, Keegan JM, Mathews DH: **Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change.** *BMC Bioinformatics* 2006., **7, article 173:**

12. Weinberg Z, Ruzzo WL: **Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy.** *Bioinformatics* 2004,**20**(supplement 1):i334-i341.

13. Weinberg Z, Ruzzo WL: **Sequence-based heuristics for faster annotation of non-coding RNA families.** *Bioinformatics* 2006,**22**(1):35-39. 10.1093/bioinformatics/bti743

14. Yoon B-J, Vaidyanathan PP: **Fast search of sequences with complex symbol correlations using profile context-sensitive HMMS and pre-screening filters. 1:** I345-I348., Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07), Honolulu, Hawaii, USA, April 2007

15. Yoon B-J, Vaidyanathan PP: **Fast structural similarity search of noncoding RNAs based on matched filtering of stem patterns.** Proceedings of the 41st Asilomar Conference on Signals, Systems and Computers (ACSSC '07), Pacific Grove, Calif, USA, November 2007, 44-48.

16. Yoon B-J, Vaidyanathan PP: **Profile context-sensitive HMMS for probabilistic modeling of sequences with complex correlations. 3:**, Proceedings of 31st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06), Toulouse, France, May 2006 317-320.

17. Yoon B-J, Vaidyanathan PP: **Structural alignment of RNAs using profile-csHMMs and its application to RNA homology search: overview and new results.** *IEEE Transactions on Automatic Control* 2008, **53:** 10-25.

18. Nawrocki EP, Eddy SR: **Query-dependent banding (QDB) for faster RNA similarity searches.** *PLoS Computational Biology* 2007,**3**(3):e56. 10.1371/journal.pcbi.0030056

19. Yoon B-J, Vaidyanathan PP: **Fast structural alignment of RNAs by optimizing the adjoining order of profile-cs HMMs.** *IEEE Journal on Selected Topics in Signal Processing* 2008,**2**(3):400-411.

20. Matsui H, Sato K, Sakakibara Y: **Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures.** *Bioinformatics* 2005,**21**(11):2611-2617. 10.1093/bioinformatics/bti385

21. Rabiner LR: **A tutorial on hidden Markov models and selected applications in speech recognition.** *Proceedings of the IEEE* 1989,**77**(2):257-286. 10.1109/5.18626

22. Yoon B-J, Vaidyanathan PP: **Context-sensitive hidden Markov models for modeling long-range dependencies in symbol sequences.** *IEEE Transactions on Signal Processing* 2006,**54**(11):4169-4184.

23. Klein RJ, Eddy SR: **RSEARCH: finding homologs of single structured RNA sequences.** *BMC Bioinformatics* 2003., **4, article 44:**

24. Gorodkin J, Heyer LJ, Stormo GD: **Finding the most significant common sequence and structure motifs in a set of RNA sequences.** *Nucleic Acids Research* 1997,**25**(18):3724-3732. 10.1093/nar/25.18.3724

25. Griffiths-Jones S, Moxon S, Marshall M, Khanna A, Eddy SR, Bateman A: **Rfam: annotating non-coding RNAs in complete genomes.** *Nucleic Acids Research* 2005, **33, database issue:** D121-D124.

26. Rivas E, Eddy SR: **The language of RNA: a formal grammar that includes pseudoknots.** *Bioinformatics* 2000,**16**(4):334-340. 10.1093/bioinformatics/16.4.334

This PDF was created after Publication.