A Bayesian approach to pairwise RNA Secondary Structure Alignment

Zhuozhi Wang and Elisabeth R. M. Tillier
Cancer Genomics & Proteomics, Ontario Cancer Institute,
University Health Network, Room 5-354, 101 College St.
Toronto, Ontario M5G 1L7 Canada
zwang@uhnres.utoronto.ca
e.tillier@utoronto.ca

Abstract

In solving the question of alignment of two RNA secondary structures, it is not clear how to choose parameters for the introduction of gaps in the alignment. We introduce a Bayesian approach for aligning two RNA secondary structures. The algorithm can not only compute the optimal alignment itself, but also some interesting posteriors such as posteriors of substitution matrix and gap opening penalties. The running time of this algorithm is $O(p_1 p_2 n m)$, where $p_1$ and $p_2$ are the number of base pairs in RNA structures $A$ and $B$, and $n$ and $m$ are the lengths of the two RNA secondary structures respectively.

1. Introduction

Ribonucleic Acid (RNA) sequences have been shown to play numerous important biological roles: transferring genetic information, regulating gene expression, and enzymatic catalysis. They are single-stranded polymers which form higher-order structures with intra-strand base pairs. It is this higher-order structure that determines how the RNA sequences play their roles. Contiguous base-pairs form stems. And depending on the number and position of single bases, several RNA secondary substructures are possible: hairpin loops, bulges, and multiple loops. The secondary structure of RNA is formed with such substructures. Tertiary structure of RNA consists of more complicated substructures: pseudoknots when base-pairings occur between the loops and triples when one base pairs with more than one bases.

Since like protein sequences, it is the structure that provides functionality, the RNA structure remains conserved whereas the RNA sequences in a family may be very diverged. Therefore, to determine the similarity between two RNA sequences, it is better to compare their structures then their sequences. RNA secondary structures are suitable for this purpose, since they contain most of the structural information, namely base pairs.

The question of aligning two RNA structures has been studied extensively. The current methods used can be roughly classified into three different categories. The first approach treats RNA secondary structure as strings [2] [5] [11] [12] [14] [20]. Specifically, these algorithms treat RNA structures as strings with some characters connected by arcs. The algorithms usually start with short substrings, and then extend to the whole string using those pre-computed sub-alignments. The second approach involves treating the RNA secondary structures as trees [1] [4] [8] [16] [17]. These algorithms consider that RNA secondary structures are tree-shaped, and thus the problem of comparing two RNA secondary structures becomes that of comparing two trees. The third approach uses a probability model, such as those using Stochastic context free grammar [10]. The authors defined a set of grammar productions, each of which has a certain probability, the optimal alignment of the two structures is the one has the maximum probability derived from the productions.

Bayesian based methods have been widely used in Bioinformatics research, such as in sequence analysis [22], RNA structure prediction [6] and phylogenetic analysis [9]. One advantage of Bayesian based methods is that they provide a confidence estimate of the results. For example, for the problem of pairwise sequence alignment, the Bayesian based method can be used to compute the likelihood of the alignment of two amino acids under some specified conditions [22].

In this paper, we will present a Bayesian based algorithm to compute the alignment of pairwise RNA secondary structures and some useful posterior probabilities.
2. Methods

An RNA sequence is generally considered as a sequence of characters over the letter set \{A, C, G, U\}. Base pairings occur between A and U, G and C, or G and U. The left side of an RNA sequence is called \(3'\) end, while the right side \(3'\) end. In this section, we will briefly introduce the concept of RNA secondary structure and the edit operations on RNA structure. We will also introduce basic concepts about Bayesian inference and set up the Bayesian model of aligning pairwise RNA secondary structures.

2.1. RNA secondary structure and operations on RNA structure

We define two RNA secondary structures \(A\) and \(B\). Additionally, \(A_i\) represents the \(i\)th base in structure \(A\), and \((A_i, A_j)\) represents a base pair (if the context is clear, we only use \((i, j)\), \(A_{i:j}\) to represent the bases in \(A\) indexed from \(i\) to \(j\) inclusively, and \(|A|\) the length of \(A\).

Concept 1: (RNA Secondary Structure) The secondary structure of an RNA sequence \(A\) is a set of base pairs, say \(S\), which should satisfy the following conditions:

1. For any base pair \((A_i, A_j)\), it must be one of the canonical base pairs, \(A-U\) (\(U-A\), \(G-C\) (\(C-G\)) or \(G-U\) (\(U-G\)).
2. For any two base pairs \((A_{i_1}, A_{j_1})\) and \((A_{i_2}, A_{j_2})\), either \(i_1 = i_2\) and \(j_1 = j_2\) or \(i_1 \neq i_2\) and \(j_1 \neq j_2\).
3. If \(h < i < j < k\), then \(S\) cannot contain both \((A_h, A_j)\) and \((A_i, A_k)\).
4. If \(S\) contains \((A_i, A_j)\), then \(|j - i| \geq 4\).

In an RNA secondary structure, if \(A_i\) pairs with \(A_j\), we define \(p(i) = j\) and \(p(j) = i\). The smaller of the two values \(i\) and \(j\) is called the \(3'\) end of a base pair, the larger one is called the \(5'\) end. If \(A_i\) is a single base (it does not pair with any other bases), then we define \(p(i) = i\).

We consider the RNA base pairs as entities upon which edit operations can be performed. We adopt the computational model introduced in [20], in which the set of edit operations [15] is extended to allow operations upon the RNA base pairs. We thus have single base insertion, deletion and substitution, and base pair insertion, deletion and substitution. When a deletion or an insertion is applied to a base pair, both of the bases consisting of the base pair are deleted or inserted. Substitutions refer to the alignment of base or base-pairs and may or may not involve different bases (the bases can match). A substitution of a base pair results in a substitution of both of its bases. Figure 1 shows the possible operations applied to an RNA secondary structure alignment.

Concept 2: (Global RNA Secondary Structure Alignment) Given two RNA secondary structures \(A\) and \(B\), we define the alignment \((A', B')\) of \(A\) and \(B\) as following:

1. \(A'\) is \(A\) and \(B'\) is \(B\) except some spaces (‘-’) are inserted and \(|A'| = |B'|\).
2. If \(A'_i\) is a single base in \(A'\), then either \(B'_i\) is a single base or a space (‘-’); and vice versa.
3. If \((A'_i, A'_j)\) is a base pair in \(A'\), then either \((B'_i, B'_j)\) is a base pair or both of them are spaces (‘-’); and vice versa.
4. There are no opposing spaces in the alignment, e.g. - - -.

2.2. Bayesian Inference

Bayesian inference uses a probabilistic approach. As reviewed in [3], there are standard steps to make Bayesian inference from a given data set. First a model is defined which expresses qualitative aspects of our knowledge (e.g., forms of distributions, independence assumptions). Generally, the model has some unknown parameters. We then need to specify a prior probability distribution for these unknown parameters that expresses our beliefs about which values are more or less likely. All those known or unknown data are incorporated into a joint probability model. Next, we have to compute those posterior distributions conditional on the observed data. Last, the model can be evaluated by considering its fit to the data. For our purpose, we only concentrate on the first three steps.

Following the convention in book [3], we use \(y\) to denote the observed data, and \(\theta\) the unobserved parameters, then we have

\[
p(y, \theta) = p(y|\theta) \times p(\theta)
\]

and

\[
p(y, \theta) = p(\theta|y) \times p(y).
\]

Usually, we are interested in the so-called posterior distribution \(p(\theta|y)\), which tells us what is known about the parameters \(\theta\) given the data \(y\). By the preceding two equations, we have

\[
p(\theta|y) = \frac{p(\theta,y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{1}
\]

Therefore, if the prior probability \(p(y)\) is specified, \(p(\theta|y)\) is proportional to the multiplication of \(p(y|\theta)\) and \(p(\theta)\). The value of \(p(\theta|y)\) can thus be computed by integrating the likelihood of the data with respect to the parameters in the model.
3. Application to the alignment of two RNA secondary structures

To accomplish the Bayesian analysis, we adopted the methods that have been widely used in [13] [19] [22], especially we have adopted the gap-model discussed by Liu et al in [13]. In our application, we consider RNA structures A and B, which have lengths m and n respectively, as observed data. A series of matrices $M$ analogous to scoring matrices, such as BLOSUM[7], are parameters. $G$ is a set of pairs consisting of gap opening $g_o$ and gap extension $g_e$ ratios. We view $S$, the alignment of $A$ and $B$, as a matrix whose values are either 1 or 0. In $S$, for any $j$, where $B_j$ is a single base, $\sum_{i=1}^{n} S_{ij} \leq 1$; and for any $i$ where $A_i$ is a single base, $\sum_{j=1}^{m} S_{ij} \leq 1$. For a base pair $A_{i_1}B_{j_1}$, either $\sum_{j_1=1}^{m} S_{i_1j_1} = 1$ and $\sum_{j_2=1}^{m} S_{i_2j_2} = 1$, or $\sum_{j_1=1}^{n} S_{i_1j_1} = 0$ and $\sum_{j_2=1}^{n} S_{i_2j_2} = 0$.

Similarly, for a base pair $B_{j_1}A_{i_2}$, either $\sum_{i_1=1}^{m} S_{ij_1} = 1$ and $\sum_{i_2=1}^{m} S_{ij_2} = 1$, or $\sum_{i_1=1}^{n} S_{ij_1} = 0$ and $\sum_{i_2=1}^{n} S_{ij_2} = 0$. These formulas mean that each single base (or base pair) in $A$ is (are) aligned with at most one base (base pair) in $B$ and a base pair cannot be broken; and vice versa. Even though we call $S$ an alignment of $A$ and $B$, yet bear in mind, we do not know which bases are involved in the RNA structures.

3.1. The joint probability model

In previous papers, such as [2] [11] [20] [21], given a scoring matrix $M_a$ and a specific set of gap parameters $G_a$, an alignment is obtained that has the maximum probability:

$$\max_S \{ P(A, B|S, G_a, M_a) \times P(S|G_a) \}. \quad (2)$$

In the Bayesian model, we avoid using a specific $M_a$ and $G_a$, instead, we integrate over a set of possible gap penalty values and a series of scoring matrices. For this RNA secondary structure alignment problem, we set up the joint distribution, $Joint = likelihood \times priors$, as following:

$$P(A, B, S, M, G) = P(A, B|S, M, G) \times P(S|M, G) \times P(M) \times P(G). \quad (3)$$

Note that in this model, we do not have separate matrices for single bases and base pairs, we assume that $M$ specifies probabilities for both of them—single bases vs. single bases and base pairs vs. base pairs. Furthermore, in $M$ there are no values for single base substituted by base pairs. We assume that given $M$, $S$ and $G$, the joint probability of $A$ and $B$ is independent of $G$. Given $M$ and $G$, prior to seeing the RNA structures, $M$ is independent of $S$. Finally, we assume that $M$ is independent of $G$. Therefore Equation 3 can be rewritten as following:

$$P(A, B, S, M, G) = P(A, B|S, M) \times P(S|G) \times P(M) \times P(G). \quad (4)$$

3.2. The Likelihood

For a specific scoring matrix $\gamma$, the scoring value for a substitution $A_i$ with $B_j$ is calculated as following (ignoring the scaling constant):

$$\gamma(A_i, B_j) = \log \frac{M(A_i, B_j)}{\mathcal{M}(\cdot, B_j)} \quad (5)$$

where $\mathcal{M}(\cdot, B_j)$ and $\mathcal{M}(A_i, \cdot)$ are the marginal distribution of $A_i$ and $B_j$ respectively, and $\mathcal{M}(A_i, B_j)$ is the joint probability of aligned bases $A_i$ and $B_j$.

Similarly, for the substitution of base pair $(A_{i_1}, A_{i_2})$ with $(B_{j_1}, B_{j_2})$, the score is calculated as following [18]:

$$\gamma((A_{i_1}, A_{i_2}); (B_{j_1}, B_{j_2})) = \log \frac{\mathcal{M}((A_{i_1}, A_{i_2}); (B_{j_1}, B_{j_2}))}{\mathcal{M}((A_{i_1}, A_{i_2}); \cdot) \mathcal{M}(\cdot, (B_{j_1}, B_{j_2}))} \quad (6)$$
where \( \mathcal{M}(\{A_i, A_{i_2}\}; \cdot) \) and \( \mathcal{M}(\cdot, \{B_j, B_{j_2}\}) \) are the marginal distribution of base pairs \( \{A_i, A_{i_2}\} \) and \( \{B_j, B_{j_2}\} \) respectively, and \( \mathcal{M}(\{A_i, A_{i_2}\}; \{B_j, B_{j_2}\}) \) is the joint probability of aligned base pairs \( \{A_i, A_{i_2}\} \) and \( \{B_j, B_{j_2}\} \). Therefore the joint probability of \( A \) and \( B \), given \( \mathcal{M} \) and \( S \), can be written as,

\[
\log P(A, B|S, \mathcal{M}) = \sum S_{ij} \times \gamma(A_i, B_j) + \sum \mathcal{M}(A_i; \cdot) + \sum \mathcal{M}(\cdot, B_j) + \sum S_{i_2, i_2, j_2, j_2} \times r + \sum \mathcal{M}(\{A_i, A_{i_2}\}; \cdot) + \sum \mathcal{M}(\cdot; \{B_j, B_{j_2}\})
\]

(7)

where \( r = \gamma((A_i, A_{i_2}), (B_j, B_{j_2})) \). In Equations 5, 6 and 7, we can treat the marginal probabilities, \( \mathcal{M}(A_i; \cdot) \), \( \mathcal{M}(\cdot, B_j) \), \( \mathcal{M}(\{A_i, A_{i_2}\}; \cdot) \) and \( \mathcal{M}(\cdot; \{B_j, B_{j_2}\}) \), as 1. Thus the joint probability of \( A \) and \( B \), \( P(A, B|S, \mathcal{M}) \), is proportional only to the exponents of elements of scoring matrices \( \gamma \). In later explanation, we will not distinguish matrices \( \mathcal{M} \) and \( \gamma \) and ignore the contributions of \( \mathcal{M}(A_i; \cdot) \), \( \mathcal{M}(\cdot, B_j) \), \( \mathcal{M}(\{A_i, A_{i_2}\}; \cdot) \) and \( \mathcal{M}(\cdot; \{B_j, B_{j_2}\}) \).

### 3.3. The Priors

In Equation 4, there are several priors we need to consider. We assume that the scoring-matrix-like \( \mathcal{M} \) is independent of \( G \) a priori, and the prior probability of \( S \) is independent of \( \mathcal{M} \) given \( G \). We assume non-informative priors on \( \mathcal{M} \) and \( G \). This means \( p(\mathcal{M}) = \frac{1}{N_{\mathcal{M}}} \), where \( N_{\mathcal{M}} \) is the total number of matrices; and \( p(G) = \frac{1}{N_{G}} \), where \( N_{G} \) is the total number of gap penalty parameters.

For a given \( S^* \), the \( P(S^*|G) \) can be calculated by the following:

\[
P(S^*|G) = \frac{N_{G}(S^*) L_{G}(S^*)}{\sum S_{G} N_{G}(S) L_{G}(S)}
\]

(8)

where \( G \) consists of \( g_r \) and \( g_o \), \( N_{G}(S) \) is the number of gaps in \( S \), and \( L_{G}(S) \) is the total length of the gaps in \( S \). Note that \( S^* \) does not contain any base information. Thus its prior probability solely depends upon the gap information.

### 3.4. Posteriors

The posteriors we are interested in are the conditional probabilities of \( \mathcal{M} \) and \( G \). Since with those parameters determined, we could draw the alignment of the given two structures \( A \) and \( B \). We derive these posterior probabilities by integrating over all the other parameters in the model (namely \( \mathcal{M} \) or \( G \)) and all possible alignments \( S \):

\[
P(G|A, B) = \sum_{\mathcal{M}} \sum_{G} P(A, B|\mathcal{M}, S) P(S|G) P(G) P(\mathcal{M})
\]

\[
P(\mathcal{M}'|A, B) = \sum_{\mathcal{M}} \sum_{G} P(A, B|\mathcal{M}', S) P(S|G) P(G) P(\mathcal{M}')
\]

Equations 9 and 10 can be further simplified as following, since \( P(G) = P(G') \) and \( P(\mathcal{M}) = P(\mathcal{M}') \):

\[
P(G'|A, B) = \frac{\sum \sum P(A, B|\mathcal{M}', S) P(S|G')}{\sum \sum P(A, B|\mathcal{M}, S) P(S|G)}
\]

(11)

\[
P(\mathcal{M}'|A, B) = \frac{\sum \sum P(A, B|\mathcal{M}', S) P(S|G)}{\sum \sum P(A, B|\mathcal{M}, S) P(S|G)}
\]

(12)

Given a specific alignment \( S^* \), it is interesting to know how likely the alignment can be generated given a set of matrices and gap penalties. The posterior probability of a given alignment \( S^* \) can be calculated by the following:

\[
P(S^*|A, B) = \frac{\sum \sum P(A, B|S^*, \mathcal{M}) P(S^*|G)}{\sum \sum P(A, B|S, \mathcal{M}) P(S|G)}
\]

(13)

### 4. Algorithms

To finish calculating the posterior probabilities described above, it requires us to compute the denominators and numerators of both Equations 11 and 12. A further examination reveals that we only need to compute: \( \sum_{\mathcal{M}} \sum_{G} P(A, B|\mathcal{M}, S) P(S|G) \). By using Equation 8, we have

\[
\sum_{\mathcal{M}} \sum_{G} P(A, B|\mathcal{M}, S) P(S|G) = \frac{\sum_{\mathcal{M}} \sum_{G} P(A, B|\mathcal{M}) g_r N_{G}(S) L_{G}(S)}{\sum_{\mathcal{M}} \sum_{G} g_r N_{G}(S) L_{G}(S)}
\]

(14)

It is impossible to enumerate all the possible alignments \( S \) to complete the summation. In this section, we introduce dynamic programming algorithms for calculating the summation. We will also present an algorithm for sampling structures in exact proportion to their posterior probabilities. The summation algorithms are analogous to those proposed in [20].

Our algorithms are bottom-up dynamic programming algorithms. We compute the summations of all pairs of small pieces of structures from \( A_{i_1, i_2} \) and \( B_{j_1, j_2} \). The small
pieces are then combined to obtain the necessary summations of $A$ and $B$.

### 4.1. Calculating the Denominator of Equation 14

We first describe an algorithm to compute the gap probability of alignment between two substructures $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$, where $A_{i_1-1}$ and $A_{i_2+1}$ are paired, and so are $B_{j_1-1}$ and $B_{j_2+1}$. Let $ND(i_1, i; j_1, j)$ be the gap probability of alignments between $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$, with $A_i$ and $B_j$ aligned with each other; $NV(i_1, i; j_1, j)$ the probability of alignments between $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$ with $A_i$ deleted; and $NH(i_1, i; j_1, j)$ be the probability alignments between $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$ with $B_j$ inserted. We use $NT(i_1, i; j_1, j)$ to denote the sum of the probabilities of alignments between $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$. We impose the restriction that a deletion in $A$ cannot be followed by an insertion in $B$. This prevents us from counting an alignment twice.

When a deletion or an insertion occurs at $A_i$ or $B_j$, we then have the following equations,

\[
NV(i_1, i; j_1, j) = (NV(i_1, i - 1; j_1, j) + ND(i_1, i - 1; j_1, j) \times g_o) \times g_o \quad (15)
\]

\[
NH(i_1, i; j_1, j) = (NH(i_1, i; j_1 - 1) + ND(i_1, i; j_1 - 1) \times g_o + NV(i_1, i; j_1 - 1) \times g_o) \quad (16)
\]

Note that for an insertion/deletion, we do not distinguish if $A_i$ or $B_j$ is a single base or a part of base pair. It is not hard to see that if $A_i$ or $B_j$ is part of a base pair, then both of the bases are deleted/inserted. Since the only place where two base pairs are aligned with each other is in case 2 of calculating $ND(i_1, i; j_1, j)$. Due to the restriction that a deletion cannot be followed by an insertion, the calculation of $NV$ uses two terms, while the one of $NH$ uses three.

For $ND(i_1, i; j_1, j)$, we need to consider three cases depending on whether $A_i$ and $B_j$ are single bases or base pairs:

1. Both $A_i$ and $B_j$ are single bases of $A$ and $B$.

\[
ND(i_1, i; j_1, j) = NH(i_1, i - 1; j_1, j - 1) + ND(i_1, i - 1; j_1, j - 1) + NV(i_1, i - 1; j_1, j - 1) \quad (17)
\]

Before the current substitution site $A_i$ and $B_j$, there are three possibilities: insertion, deletion and substitution. We need to sum over these possibilities.

2. Both $A_i$ and $B_j$ are 3’ end of a base pair in $A$ and $B$ respectively, and $p(i) \leq i_1$ and $p(j) \leq j_1$. Namely base pairs $(A_{p(i)}, A_i)$ and $(B_{p(j)}, B_j)$ are in substructures $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$ respectively.

\[
ND(i_1, i; j_1, j) = NT(i_1, i; j_1, j) \times g_o \quad (18)
\]

where $NT(i_1, i; j_1, j)$ is the summation of $NV$, $ND$ and $NH$, such as The total number of the alignments of substructures $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$ is the summation of $NV$, $NH$ and $ND$:

\[
NT(i_1, i; j_1, j) = NV(i_1, i; j_1, j) + NH(i_1, i; j_1, j) + ND(i_1, i; j_1, j) \quad (19)
\]

The alignment between base pairs $(A_{p(i)}, A_i)$ and $(B_{p(j)}, B_j)$ breaks the alignment between $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$ into three parts: alignment of $A_{[i_1:i_2 - 1]}$ and $B_{[j_1:j_2 - 1]}$, alignment of the base pairs $(A_{p(i)}, A_i)$ and $(B_{p(j)}, B_j)$, and alignment of $A_{[i_1:i_2]}$ and $B_{[j_1:j_2 + 1]}$. The multiplication of the appropriate summations gives us the total number of probabilities between $A_{[i_1:i_2]}$ and $B_{[j_1:j_2]}$ ending with $A_i$ and $B_j$ aligned. Note that this is the only case where two base pairs are aligned with each other.

3. For other cases, we have

\[
NV(i_1, i; j_1, j) = 0. \quad (20)
\]

There are several sub-cases we need to consider:

(a) $A_i$ is 3’ end, but $B_j$ is not, according to rules 1 and 2, both $A_i$ and $B_j$ must be deleted, since we do not allow $B_j$ to be aligned with $A_i$.

(b) $A_i$ is 5’ end, at this point, we defer the consideration of alignment of $A_i$ to the time when find its 3’-end correspondence.

(c) $B_j$ is 3’ end, but $A_i$ is not, this is similar to the first sub-case.

(d) $B_j$ is 5’ end, this is similar to the second sub-case.

Let $\phi$ be an empty structure. For any $i_1 \leq i \leq i_2$ and $j_1 \leq j \leq j_2$, the initial conditions are:

\[
NV(\phi; \phi) = g_o \quad (21)
\]

\[
NH(\phi; \phi) = g_o \quad (22)
\]

\[
ND(\phi; \phi) = 0 \quad (23)
\]

\[
NV(i_1, i; \phi) = g_o \times NV(i_1, i - 1; \phi) \quad (24)
\]

\[
NH(i_1, i; \phi) = 0 \quad (25)
\]

\[
ND(i_1, i; \phi) = 0 \quad (26)
\]

\[
NH(\phi; j_1, j) = g_o \times NH(\phi; j_1, j - 1) \quad (27)
\]

\[
NV(\phi; j_1, j) = 0 \quad (28)
\]

\[
ND(\phi; j_1, j) = 0 \quad (29)
\]

We set $NT(\phi; \phi) = 1$. $NV(\phi; \phi)$ and $NH(\phi; \phi)$ are set $g_o$ so that it is easy for us to write recurrences. We
now describe our algorithm to compute the total number of alignments between $A_{1,m}$ and $B_{1,n}$. First we assume that there are two virtual base pairs $(A_0,A_{m+1})$ and $(B_0,B_{n+1})$ which do not contribute to the probability of the alignment.

for each base pair $p$ from $A$
  for each base pair $q$ from $B$
    compute $NV(p_1+1,p_r-1;q_1+1,q_r-1)$,
    $NH(p_1+1,p_r-1;q_1+1,q_r-1)$, and
    $ND(p_1+1,p_r-1;q_1+1,q_r-1)$ arrays and store the values
  compute $NV$, $NH$ and $ND$ for base pair $(A_0,A_{m+1})$
  and $(B_0,B_{n+1})$

where $p_1$ and $p_r$ are respectively the $3'$ and $5'$ indices of base pair $p$. The time complexity of this algorithm is approximately $O(p_1p_2mn)$, where $p_1$ and $p_2$ are the total number of base pairs in $A$ and $B$, and $m$ and $n$ are the lengths of $A$ and $B$ respectively. The space complexity of the algorithm is $O(mn)$.

### 4.2. Calculating the Numerator of Equation 14

In this section, we present an algorithm to compute the summation of the probabilities for the alignment of substructures $A_{[i_1,i_2]}$ and $B_{[j_1,j_2]}$. The framework and computational model of this algorithm are the same as above. We will restrict the computation to those substructures $A_{[i_1,i_2]}$ and $B_{[j_1,j_2]}$, where $A_{[i_1-1,i_2+1]}$ and $B_{[j_1-1,j_2+1]}$ are base pairs.

Let $PD(i_1,i_2;j_1,j_2)$ denote the partial summation of probability of alignments between $A_{[i_1,i_2]}$ and $B_{[j_1,j_2]}$ and $A_1$ and $B_1$ aligned with each other, $PV(i_1,i_2;j_1,j_2)$ the partial summation of probabilities of alignments between $A_{[i_1,i_2]}$ and $B_{[j_1,j_2]}$ with $A_{[i]}$ deleted, and $PH(i_1,i_2;j_1,j_2)$ the partial summation of probabilities of alignments between $A_{[i_1,i_2]}$ and $B_{[j_1,j_2]}$ with $B_{[j]}$ inserted.

For notational convenience, we use $PT(i_1,i_2;j_1,j_2)$ to denote the total of the probabilities of all the alignments between $A_{[i_1,i_2]}$ and $B_{[j_1,j_2]}$:

$$PT(i_1,i_2;j_1,j_2) = PD(i_1,i_2;j_1,j_2) + PV(i_1,i_2;j_1,j_2) + PH(i_1,i_2;j_1,j_2)$$  \hspace{1cm} (30)

In the alignment of $A_{[i_1,i]}$ with $B_{[j_1,j]}$, if a deletion or an insertion occurs at $A_{[i]}$ or at $B_{[j]}$, we have,

$$PV(i_1,i_2;j_1,j_2) = (PV(i_1,i_2-1;j_1,j_2) + PD(i_1,i_2-1;j_1,j_2) \times g_o) \times g_b$$  \hspace{1cm} (31)

$$PH(i_1,i_2;j_1,j_2) = (PV(i_1,i_2;j_1,j_2-1) \times g_o + PD(i_1,i_2;j_1,j_2-1) \times g_o + PH(i_1,i_2;j_1,j_2-1) \times g_o$$  \hspace{1cm} (32)

The calculation of $PD(i_1,i_2;j_1,j_2)$ is similar to that of $ND(i_1,i_2;j_1,j_2)$. Again we have three cases to consider:

1. Both $A_{[i]}$ and $B_{[j]}$ are single bases;
   $$PD(i_1,i_2;j_1,j_2) = PT(i_1,i_2-1;j_1,j_2-1) \times \exp(\gamma(i,j))$$  \hspace{1cm} (33)

2. Both $A_{[i]}$ and $B_{[j]}$ are $3'$ end of base pairs;
   $$PD(i_1,i_2;j_1,j_2) = PT(i_1,i_2-1;j_1,j_2-1) \times PT(p(i_1-1,i_1-1)+p(j_1-1,j_1-1)) \times \exp(\gamma(p(i_1-1,i_1-1)+p(j_1-1,j_1-1)))$$  \hspace{1cm} (34)

3. Otherwise, we set
   $$PD(i_1,i_2;j_1,j_2) = 0.$$  \hspace{1cm} (35)

The explanations are similar to those of Equations 17, 18 and 20. We set the boundary conditions as following, for any $i_1 \leq i \leq i_2$ and $j_1 \leq j \leq j_2$,

$$PV(\phi, \phi) = g_o$$  \hspace{1cm} (36)

$$PH(\phi, \phi) = g_o$$  \hspace{1cm} (37)

$$PD(\phi, \phi) = 0$$  \hspace{1cm} (38)

$$PV(i_1,i_2-1; \phi) = PV(i_1,i_2-1; \phi) \times g_e$$  \hspace{1cm} (39)

$$PH(i_1,i_2-1; \phi) = 0$$  \hspace{1cm} (40)

$$PD(i_1,i_2-1; \phi) = 0$$  \hspace{1cm} (41)

$$PH(\phi,j_1,j_2-1) = PH(\phi,j_1,j_2-1) \times g_e$$  \hspace{1cm} (42)

$$PV(\phi,j_1,j_2-1) = 0$$  \hspace{1cm} (43)

$$PD(\phi,j_1,j_2-1) = 0$$  \hspace{1cm} (44)

Again, we set $PT(\phi, \phi) = 1$. The high level algorithm is the same as the one in the above section. The time complexity of the algorithm is $O(mn)$.

### 4.3. Sampling alignments from the exact posterior probability distribution

We now describe an algorithm to sample alignments following their exact posterior probability distribution. The process of sampling alignments in proportion to their joint posterior probability is similar to the traceback procedure used in [20]. However, because of the unknown values of the parameters $M$ and $G$, we have two additional steps. First, we need to draw a scoring matrix from $M$ by $P(M|A,B)$. Then we sample the gap penalty parameter $G$, conditional on $M$. Only after these two steps can we draw the alignment itself conditional upon $M$ and $G$.

At first, we need to draw a sample of the last pair of amino acids in two structures $A$ and $B$, namely $A_{m}$ and $B_{n}$. There are three possibilities according to the values: $PD_{(m,n)}^M$, $PV_{(m,n)}$, and $PH_{(m,n)}$. Note that it is possible that $A_{m}$ and $B_{n}$ are $3'$ end of a base pair, for which we use a different formula.
Generally, at positions $A_i$ and $B_j$ (assuming we are in the first round of sampling), we determine the operations applied at this moment by the largest value of $P_D(1:i;1:j)$ (substitution), $P_V(1:i;1:j)$ (insertion) and $P_H(1:i;1:j)$ (deletion). Thus, we have three cases:

1. If substitution is applied and both $A_i$ and $B_j$ are single bases, then we change indices as $i = i - 1$ and $j = j - 1$. If substitution is applied and both $i$ and $j$ are $3'$ end of a base pair in $A$ and $B$, we change the indices as following $i = p(i) - 1$ and $j = p(j) - 1$.

2. If deletion is applied, then we change only $i$, namely $i = i - 1$.

3. If insertion is applied, then we change only $j$, namely $j = j - 1$.

The first round of sampling returns us the outer-most base pairs and single bases. The sampling algorithm then traces back into each pair of aligned base pairs. The whole sampling process repeats recursively until all the indices of $A$ and $B$ have been processed. The time complexity of the sampling process is $O(\max\{m, n\})$.

4.4. The optimal alignment

The optimal alignment is the one that achieves the maximum for the joint probability: $P(A, B, S, G, M)$. After the computation of posteriors of $G$ and $M$, we can determine the best set of $G$ and $M$ given $A$ and $B$. Thus, by using the back-tracing algorithm in second 4.3, we find the optimal alignment of of the $A$ and $B$.

4.5. Calculating the alignment of two specific bases (base pairs)

One of the advantages using a Bayesian approach is the ability to determine the probability of each pair of bases are aligned. Let $S_{ij}$ denote an alignment in which single bases $A_i$ and $B_j$ are aligned.

$$P(S_{ij} = 1|A, B) = \frac{\sum \sum \sum P(A, B|G')P(G')P(M')P(G)}{\sum \sum \sum P(A, B|G')P(G')P(M')P(G)}$$

(45)

The numerator can be calculated by using the algorithm in Section 4.2, except that we fix that $A_i$ and $B_j$ must be aligned. Similarly, we could calculate the probability of aligning two base pairs $(A_{p(i)}, A_i)$ and $(B_{p(j)}, B_j)$.

5. Conclusion

In this paper, we have introduced several Bayesian-based algorithms to calculate the interesting posteriors of an alignment $S$ of two RNA secondary structures $A$ and $B$. The Bayesian model is based upon the one introduced in [13]. The Bayesian model has provided us a way to go over a set of gap parameters and a series of scoring matrices instead of just one. In the future, it may be interesting to include base pair break operation in the computation model.

References


