NAIST-IS-DD0361204

Doctoral Dissertation

Multiple Alignment for Structural RNA Sequences

Hisanori Kiryu

March 13, 2007

Department of Bioinformatics and Genomics
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of SCIENCE

Hisanori Kiryu

Thesis Committee:

Professor Shunsuke Uemura, (Supervisor)
Professor Naotake Ogasawara, (Co-supervisor)
Professor Shigehiko Kanaya, (Member)
Professor Kiyoshi Asai, (Member)

Multiple Alignment for Structural RNA Sequences*

Hisanori Kiryu

Abstract

Recent transcriptomic studies have revealed the existence of a considerable number of non protein coding RNA transcripts in higher eukaryotic cells. To investigate the functional roles of those transcripts, it is of great importance to find conserved secondary structures from multiple alignments at a genomic scale. In this thesis, I investigate the problem of multiple alignment of structural RNA sequences.

The first part of the thesis presents a novel alignment algorithm for structural RNA sequences. Structural RNA genes show unique evolutionary conservation patterns to conserve their secondary structures, which should be taken into account for constructing accurate multiple alignments of RNA genes. An algorithm that naturally includes the base covariance effect in its alignment model is introduced which has an efficient scoring system that considerably reduces the time and space requirement without degradation of the alignment quality. Sevaral experiments are performed to show that the alignment quality and the accuracy of the consensus secondary structure prediction from the alignment are the highest among the leading alignment programs. The algorithm can align relatively long RNA sequences such as eukaryotic type signal recognition particle RNA of length about 300 nucleotides which has not been computable by other Sankoff-based algorithms.

The second part of the thesis presents novel algorithms that predict consensus secondary structures from the multiple alignment of RNA sequences. I compare

*Doctoral Dissertation, Department of Bioinformatics and Genomics, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0361204, March 13, 2007.

several algorithms for the consensus structure prediction under different levels of alignment qualities. One of these algorithms, termed McCaskill-MEA, is shown to be the robustest against alignment failures than the other algorithm. The McCaskill-MEA method performs better than other algorithms, especially when the alignment quality is low and when the alignment consists of many sequences.

Keywords:

RNA, sequence analysis, multiple alignment, maximal expected accuracy, McCaskill algorithm

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, a number of studies have shown that there are a substantial number of RNA transcripts which do not code protein sequences in higher eukaryotic cells [34],[7],[2], and the question whether such transcripts have any functional roles in cellular processes has attracted much interest. To investigate the functional roles of such transcripts, it is of great importance to find conserved secondary structures from multiple alignments created at a genomic scale.

Since structural RNA genes show unique evolutionary conservation patterns to conserve their secondary structures, multiple alignment methods should take into account the base covariance effect to construct the accurate multiple alignments. The Sankoff algorithm [38] is the alignment algorithm which naturally includes the base pair covariation effect in the alignment model. However, it was not practical to use the Sankoff algorithm for aligning multiple sequences due to its prohibitive cost of computation. Therefore, most of the studies has used alignment programs which neglect the special conservation patterns of secondary structures to search for conserved secondary structures [42], [20],[44], [43], [35]. The neglect of the base pair covariance effect has potential risks to overlook conserved secondary structures due to misalignment around the stem regions. Such loss of sensitivity is particularly problematic in the early stage of large scale screening which precedes the time-consuming but accurate experimental validation stages.

In this thesis, I propose a practical method to align multiple RNA sequences based on the Sankoff algorithm and a method that predicts conserved secondary

structures from alignments, which is robust against alignment failures.

The first part of the thesis presents a novel alignment algorithm for structural RNA sequences. An algorithm based on the Sankoff model is introduced which has an efficient scoring system that considerably reduces the time and space requirement without degradation of the alignment quality. The algorithm first compute the match probability matrix that measures the alignability of each position pair between sequences and the base pairing probability matrices for each sequence. Then these probabilities are combined to score the alignment by the Sankoff algorithm. Sevaral experiments are performed to show that the alignment quality and the accuracy of the consensus secondary structure prediction from the alignment are the highest among the leading alignment programs. The algorithm can align relatively long RNA sequences such as eukaryotic type signal recognition particle RNA of length about 300 nucleotides which has not been computable by other Sankoff-based algorithms. The algorithm is implemented as the software "Murlet".

The second part of the thesis presents novel algorithms that predict consensus secondary structures from the multiple alignment of RNA sequences. All three algorithms maximize the expected accuracy of secondary structures under different base pairing probability distributions. One of the algorithms, termed McCaskill-MEA, is shown to be the robustest against alignment failures than the other two algorithms and also the algorithms frequently used for the conserved structure prediction. The McCaskill-MEA method first computes the base pairing probability matrices for all sequences in the alignment, and then obtains the base pairing probability matrix of the alignment by averaging over those matrices. The consensus secondary structure is predicted from that matrix so that the expected accuracy of the prediction is maximized. The McCaskill-MEA method performs better than others, especially when the alignment quality is low and when the alignment consists of many sequences. The model has a parameter that controls the sensitivity and specificity of predictions, which is useful for multistep screening procedures to search for conserved secondary structures, and for assigning confidence values to the predicted base pairs.

Chapter 2

Maximal Expected Accuracy Algorithm

2.1. Overview

In this chapter, I give a few formal definitions of the secondary structure prediction and the traditional pairwise sequence alignment. I also describe the maximal expected accuracy (MEA) algorithm that forms the basis of the algorithms proposed in the following chapters.

2.1.1 Secondary Structure Prediction Problem

For a sequence x of length L, let $\mathcal{C} = \{i | 1 \leq i \leq L\}$ be the set of sequence positions and let $\mathcal{PC} = \{(i,j) \in \mathcal{C} \times \mathcal{C} | 1 \leq i < j \leq L\}$ be the set of pairs of alignment columns. A secondary structure of the sequence x is defined by the mapping $m^{(b)}$ from $\mathcal{PC}$ to the binary values $\{0,1\}$,

$$m^{(b)} : \mathcal{PC} \longrightarrow \{0,1\}$$

such that $m^{(b)}(i,j) = 1$ if the column pair (i,j) forms a base pair, and 0 otherwise. Let

$$y^{(b)} = \{y^{(b)}_{ij} \in \{0,1\} | (i,j) \in \mathcal{PC}, y^{(b)}_{ij} = m^{(b)}(i,j)\}$$

be the image of the mapping. $y^{(b)}$ cannot take all possible $2^{(L(L-1)/2)}$ values to form a consistent secondary structure. Since each column cannot be paired with two or more columns, $y^{(b)}$ satisfies the following constraint.

$$\exists(i,j), y_{ij}^{(b)} = 1 \tag{2.1}$$
$$\Longrightarrow \forall k \neq i,j, y_{ik}^{(b)} = y_{kj}^{(b)} = 0 \tag{2.2}$$

Moreover, since I do not consider pseudo-knot structures in this thesis, I assume that $y^{(b)}$ follows the nested structure constraint;

$$\begin{aligned}&\exists(i,j), y_{ij}^{(b)} = 1\\ &\Longrightarrow \forall(k,l), i < k < j < l \text{ or } k < i < l < j, y_{kl}^{(b)} = 0\end{aligned}$$

I also use an alternative representation $\mathcal{S}$ of a secondary structure, that consists of a set of loop columns $\mathcal{L}$ and a set of pair columns $\mathcal{P}$.

$$\begin{aligned}\mathcal{S} &= \{\mathcal{L}, \mathcal{P}\}\\ \mathcal{L} &= \{i \in \mathcal{C} | \forall k \neq i, y_{ik}^{(b)} = y_{ki}^{(b)} = 0\}\\ \mathcal{P} &= \{(i,j) \in \mathcal{PC} | y_{ij}^{(b)} = 1\}\end{aligned}$$

For a given conditional probability distribution $p^{(b)}(y^{(b)}|x)$, the base pairing probability $p^{(b)}(i,j)$ of columns (i,j) can be defined as follows,

$$\begin{aligned}p^{(b)}(i,j) &= E\left[\delta(y_{ij}^{(b)}, 1)\right]\\ &= \sum_{y^{(b)}} \delta(y_{ij}^{(b)}, 1) p^{(b)}(y^{(b)}|x)\\ &= \sum_{y^{(b)}|y_{ij}^{(b)}=1} p^{(b)}(y^{(b)}|x)\end{aligned}$$

where $\delta(z, z')$ is the Kronecker delta function which is defined by,

$$\delta(z,z') = \begin{cases} 1 \text{ if } z = z' \\ 0 \text{ otherwise} \end{cases}$$

and $E[A]$ is the expectation value of A with respect to $p^{(b)}(y^{(b)}|x)$. Let the loop probability $q^{(b)}(i)$ be the probability that the alignment column i does not form any pair with other columns,

$$
\begin{aligned}
q^{(b)}(i) &= E\left[\prod_{i<j}\delta(y_{ij}^{(b)},0)\prod_{j<i}\delta(y_{ji}^{(b)},0)\right] && (2.3)\\
&= E\left[\prod_{i<j}(1-\delta(y_{ij}^{(b)},1))\prod_{j<i}(1-\delta(y_{ji}^{(b)},1))\right] && (2.4)\\
&= E\left[1-\sum_{i<j}\delta(y_{ij}^{(b)},1)-\sum_{j<i}\delta(y_{ji}^{(b)},1)\right] && (2.5)\\
&= 1-\sum_{i<j}p^{(b)}(i,j)-\sum_{j<i}p^{(b)}(j,i) && (2.6)
\end{aligned}
$$

In the third line, I have used the constraint in Equation (2.2). $p^{(b)}(y^{(b)}|x)$ can be computed for various models such as models based on loop decomposition of secondary structure energy and models based on stochastic context free grammars (SCFGs). In the energy based models, $p^{(b)}(y^{(b)}|x)$ is given by the Boltzmann distribution of secondary structure configurations,

$$
\begin{aligned}
p^{(b)}(y^{(b)}|x) &= \frac{1}{Z(x)}\exp\left(-\frac{E(y^{(b)},x)}{kT}\right) && (2.7)\\
Z(x) &= \sum_{y^{(b)}}\exp\left(-\frac{E(y^{(b)},x)}{kT}\right) && (2.8)
\end{aligned}
$$

where $E(y^{(b)},x)$ is the secondary structure energy which is computed using the energy parameters collected by the Turner group [27], k is the Boltzmann constant, T is the temperature, and $Z(x)$ is the partition function. In this case, the corresponding base pairing probability matrix $p^{(b)}$ is computed by the McCaskill's algorithm [30](Algorithm 1).

The maximal likelihood prediction of the energy based models, is given by the secondary structure $y^{(b)}$ that maximizes the $p^{(b)}(y^{(b)}|x)$

$$
y^{(b)} = \mathrm{argmax}_{y^{(b)}} p(y^{(b)}|x)
$$

The Mfold algorithm [42], [46] is interpreted as one of such algorithms.

In the SCFG models [5], $p^{(b)}(y^{(b)}|x)$ is given by the sum of joint probabilities

Algorithm 1 The McCaskill Algorithm

Folding:

$$
\begin{aligned}
Q^B_{i,j} =& e^{-\mathcal{H}(i,j)/kT} + \sum_{\substack{h=i+1 \\ u \le u_{\max}}}^{j-m-2} \sum_{l=h+m+1}^{j-1} Q^B_{h,l} e^{-\mathcal{I}(i,j,k,l)/kT} + \sum_{h=i+1}^{j-m-2} Q^M_{i+1,h-1} Q^{M1}_{h,j-1} e^{-\mathcal{M}_\mathcal{C}/kT} \\
Q^{M1}_{i,j} =& \sum_{l=i+m+1}^{j} Q^B_{i,l} e^{-[\mathcal{M}_\mathcal{I}+\mathcal{M}_\mathcal{B}(j-l)]/kT} \\
Q^M_{i,j} =& \sum_{h=l+m+1}^{j-m-1} Q^M_{i,h-1} Q^{M1}_{h,j} + \sum_{h=i}^{j-m-1} Q^{M1}_{h,j} e^{-\mathcal{M}_\mathcal{B}(h-i)/kT} \\
Q^A_{i,j} =& \sum_{l=i+m+1}^{j} Q^B_{i,l} \\
Q_{i,j} =& 1 + Q^A_{i,j} + \sum_{h=i+1}^{j-m-1} Q_{i,h-1} Q^A_{h,j}
\end{aligned}
$$

Backtracking:

$$
\begin{aligned}
P^c_{h,l} =& \frac{Q_{1,h-1} Q^B_{h,l} Q_{l+1,L}}{Q_{1,L}} \\
P^i_{h,l} =& Q^B_{h,l} \sum_{\substack{i=1 \\ u<u_{\max}}}^{h-1} \sum_{j=l+1}^{L} \frac{P_{i,j}}{Q^B_{i,j}} e^{-\mathcal{I}(i,j,k,l)/kT} \\
P^m_{h,l} =& Q^B_{h,l} e^{-[\mathcal{M}_\mathcal{C}+\mathcal{M}_\mathcal{I}]/kT} \sum_{i=1}^{h-1} \left[P^{M1}_{i,l} Q^M_{i+1,h-1} + P^M_{i,l} Q^M_{i+1,h-1} + P^M_{i,l} e^{-\mathcal{M}_\mathcal{B}(h-i-1)/kT} \right] \\
P^M_{i,l} =& \sum_{j=l+2}^{L} \frac{P_{i,j}}{Q^B_{i,j}} Q^M_{l+1,j-1} \\
P^{M1}_{i,l} =& \sum_{j=l+1}^{L} \frac{P_{i,j}}{Q^B_{i,j}} e^{-\mathcal{M}_\mathcal{B}(j-l-1)/kT} \\
P_{h,l} =& P^c_{h,l} + P^i_{h,l} + P^m_{h,l}
\end{aligned}
$$

$p^{(b)}(\sigma, x)$ over the set of parses σ sharing the same secondary structure $y^{(b)}$,

$$\begin{aligned} p^{(b)}(y^{(b)}|x) &= \sum_{\sigma \in y^{(b)}} p^{(b)}(\sigma|x) \\ &= \frac{\sum_{\sigma \in y^{(b)}} p^{(b)}(\sigma, x)}{\sum_{\sigma} p^{(b)}(\sigma, x)} \end{aligned}$$

$p^{(b)}(\sigma, x)$ is the joint probability of generating the parse σ and given by the product of transition and emission probabilities of the SCFG model. The sum of the numerator in the second line is over all the parse trees which share the same secondary structure $y^{(b)}$, and the sum of the denominator is over the all the possible parse trees. The corresponding base pairing probability matrix is computed by the inside and outside algorithm [8],[4]. The computation of $p^{(b)}$ requires $\mathcal{O}(L^3)$ time and $\mathcal{O}(L^2)$ memory.

2.1.2 Maximal Expected Accuracy Algorithm

Recent studies have shown that the secondary structure predictions based on the principle of the maximization of expected accuracy (MEA) [31], perform better than the predictions made by the conventional maximal likelihood algorithm and the energy minimization algorithm [35], [4],[24], [4]. This algorithm first computes the base pairing probability $p^{(b)}(i, j)$ for each pair of alignment columns (i, j) then considers the expected accuracy $EA(\mathcal{S})$ for each secondary structure candidate $\mathcal{S}$. The predicted secondary structure $\mathcal{S}$ is obtained by maximizing the expected accuracy $EA(\mathcal{S})$ with respect to $\mathcal{S}$.

For a secondary structure $\mathcal{S} = \{\mathcal{L}, \mathcal{P}\}$ and a given parameter $\alpha^{(b)} \geq 0$, the expected accuracy $EA_{\alpha^{(b)}}(\mathcal{S})$ of $\mathcal{S}$ with respect to the conditional distribution $p^{(b)}(y^{(b)}|x)$ is defined as,

$$\begin{aligned} EA_\alpha(\mathcal{S}) &= E\left[\alpha^{(b)}\left(\sum_{i\in\mathcal{L}}\prod_{i<j}\delta(y_{ij}^{(b)}, 0) + \sum_{i\in\mathcal{L}}\prod_{j<i}\delta(y_{ji}^{(b)}, 0)\right) + 2\sum_{(i,j)\in\mathcal{P}}\delta(y_{ij}^{(b)}, 1)\right] \\ &= \alpha^{(b)}\sum_{i\in\mathcal{L}} q^{(b)}(i) + 2\sum_{(i,j)\in\mathcal{P}} p^{(b)}(i, j) \end{aligned}$$

When $\alpha^{(b)} = 1$, $EA_\alpha(\mathcal{S})$ can be interpreted as the expectation value of the number of correctly annotated bases with respect to the conditional probability distribution $p^{(b)}(y^{(b)}|x)$.

The secondary structure that maximizes the expected accuracy can be computed by the traceback procedure of Nussinov-like dynamic programming algorithm [33].

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + 2p^{(b)}(i,j) \\ M_{i-1,j} + \alpha^{(b)} q^{(b)}(i) \\ M_{i,j-1} + \alpha^{(b)} q^{(b)}(j) \\ M_{i,k} + M_{k+1,j} \text{ for } i < k < j \end{cases} \tag{2.9}$$

The maximum of the expected accuracy MEA is given by

$$\begin{aligned} \text{MEA}_{\alpha^{(b)}} &= \max_{\mathcal{S}} EA_{\alpha^{(b)}}(\mathcal{S}) \\ &= M_{1,L} \end{aligned}$$

The corresponding secondary structure $\mathcal{S} = \mathcal{S}_{\text{MEA}}$ is the MEA solution. The parameter $\alpha^{(b)}$ controls the sensitivity and specificity of the structure prediction [4]. A small $\alpha^{(b)}$ value encourages the base pair formation, which results in higher sensitivity, and a large $\alpha^{(b)}$ value encourages the increase of single stranded region and results in higher specificity .

The reason that the MEA algorithms show better prediction accuracy than their maximal likelihood counterparts can be explained as follows. In general, any computational model of secondary structure prediction based only on the sequence data has its limitation in accuracy, because an RNA molecule in reality forms a three dimensional structure in the cell, and interacts with itself among three dimensional neighborhoods of bases other than the pair forming bases. Moreover, it also interacts with bounded proteins and other cellular environments, that affect the formation of its secondary structure. Hence the absolute optimality with respect to the scoring system of the model is of limited importance. When the model is not quite correct but reasonably good, then taking the majority of structures among near optimal structures may be a more feasible way for predicting the secondary structures. The MEA algorithm can be considered as one of such algorithms.

For example, if an optimal structure does not form a pair at a column pair (i, j), but many suboptimal structures form a pair at (i, j), then the MEA solution tends to form a pair at (i, j), since the base pair probability p_{ij} takes a large value at that position.

Therefore, in contrast to the maximal likelihood method that considers the only one optimal structure, the MEA algorithm takes into account various near optimal structures and predicts the consensus structure supported by them. It presumably acts to reduce model specific artifacts from the predictions.

2.1.3 Sequence Alignment Problem

For a given pair of sequences $x^{(1)}$ and $x^{(2)}$ of lengths $L^{(1)}$ and $L^{(2)}$, an alignment between $x^{(1)}$ and $x^{(2)}$ is defined by the mapping $m^{(a)}$ from the pair positons $\mathcal{PC}^{(1,2)} = C^{(1)} \times C^{(2)}$ to the binary values $\{0, 1\}$,

$$m^{(a)} : \mathcal{PC}^{(1,2)} \longrightarrow \{0, 1\}$$

such that $m^{(a)}(i, j) = 1$ if the sequence position i of $x^{(1)}$ and the sequence position j of $x^{(2)}$ are matched in an equal column in the alignment, and $m^{(a)}(i, j) = 0$ otherwise. Let

$$y^{(a)} = \{y_{ij}^{(a)} \in \{0, 1\} | (i, j) \in \mathcal{PC}^{(1,2)}, y_{ij}^{(a)} = m^{(a)}(i, j)\}$$

be the image of the mapping. Similar to the secondary structure case, $y^{(a)}$ cannot take all possible $2^{L^{(1)} \times L^{(2)}}$ values to form a consistent alignment. Since each sequence position cannot be aligned with two or more sequence postion of the other sequence, $y^{(a)}$ satisfies the following constraint.

$$\exists (i, j), y_{ij}^{(a)} = 1 \tag{2.10}$$

$$\Longrightarrow \forall (k, l) k \neq i, l \neq j, y_{il}^{(a)} = y_{kj}^{(a)} = 0 \tag{2.11}$$

Moreover, if the positions i and j match each other in the alignment, the sequence positions left of the position i of sequence $x^{(1)}$ cannot match with the sequence postions right of the position j of sequence $x^{(2)}$ and vice versa, $y^{(a)}$ also satisfies the constraint,

$$\exists (i, j), y_{ij}^{(a)} = 1 \tag{2.12}$$

$$\Longrightarrow \forall (k, l), (i < k \text{ and } j < l) \text{ or } (k < i \text{ and } l < j), y_{kl}^{(a)} = 0 \tag{2.13}$$

We use an alternative representation $\mathcal{A}$ of the alignment, that consists of two set of insertion positions $\mathcal{I}^{(i)}$, $\mathcal{I}^{(2)}$ and a set of match pairs $\mathcal{M}$.

$$\begin{aligned}
\mathcal{A} &= \{\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \mathcal{M}\} \\
\mathcal{I}^{(1)} &= \{i \in \mathcal{C}^{(1)} | \forall l, y_{il} = 0\} \\
\mathcal{I}^{(2)} &= \{j \in \mathcal{C}^{(2)} | \forall k, y_{kj} = 0\} \\
\mathcal{M} &= \{(i,j) \in \mathcal{PC}^{(1,2)} | y_{ij}^{(a)} = 1\}
\end{aligned}$$

The match probability $p^{(a)}(i,j)$ is the posterior probability that sequence positions i and j will be matched in an alignment For a given conditional probability distribution $p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})$, the match probability $p^{(a)}(i,j)$ of a pair position (i,j) is defined by,

$$\begin{aligned}
p^{(a)}(i,j) &= E\left[\delta(y_{ij}^{(a)}, 1)\right] \\
&= \sum_{y^{(a)}} \delta(y_{ij}^{(a)}, 1) p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)}) \\
&= \sum_{y^{(a)}|y_{ij}^{(a)}=1} p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})
\end{aligned}$$

and $E[A]$ is the expectation value of A with respect to $p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})$. Let the insertion probability $q^{(a)(1)}(i)$ to be the probability that the sequence position i of sequence $x^{(1)}$ does not match with any position of the other sequence,

$$q^{(a)(1)}(i) = E\left[\prod_l \delta(y_{il}^{(a)}, 0)\right] \tag{2.14}$$

$$= E\left[\prod_l (1 - \delta(y_{il}^{(a)}, 1))\right] \tag{2.15}$$

$$= E\left[1 - \sum_l \delta(y_{il}^{(a)}, 1)\right] \tag{2.16}$$

$$= 1 - \sum_l p^{(a)}(i,l) \tag{2.17}$$

The insertion probability $q^{(a)(2)}$ of sequence $x^{(2)}$ is defined similarly. For an alignment $\mathcal{A} = \{\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \mathcal{M}\}$ and a given parameter $\alpha^{(a)} \geq 0$, the expected accuracy

$EA^{(a)}_{\alpha^{(a)}}(\mathcal{A})$ of $\mathcal{A}$ with respect to the conditional distribution $p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})$ is defined by,

$$\begin{aligned}
EA_{\alpha^{(a)}}(\mathcal{S}) &= E\left[\alpha^{(a)}\left(\sum_{i\in\mathcal{I}^{(1)}}\prod_{l}\delta(y^{(a)}_{il},0) + \sum_{j\in\mathcal{I}^{(2)}}\prod_{k}\delta(y^{(a)}_{kj},0)\right) + 2\sum_{(i,j)\in\mathcal{M}}\delta(y^{(a)}_{ij},1)\right] \\
&= \alpha^{(a)}\left(\sum_{i\in\mathcal{I}^{(1)}} q^{(a)(1)}(i) + \sum_{j\in\mathcal{I}^{(2)}} q^{(a)(2)}(j)\right) + 2\sum_{(i,j)\in\mathcal{M}} p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})
\end{aligned}$$

When $\alpha^{(a)} = 1$, $EA_{\alpha^{(a)}}(\mathcal{A})$ can be interpreted as the expectation value of the number of correctly annotated bases with respect to the conditional probability distribution $p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})$. The alignment that maximizes the expected accuracy can be computed by the traceback procedure of the following dynamic programming algorithm.

$$M_{i,j} = \max\begin{cases} M_{i-1,j-1} + 2p^{(a)}(i,j) \\ M_{i-1,j} + \alpha^{(a)}q^{(a)(1)}(i) \\ M_{i,j-1} + \alpha^{(a)}q^{(a)(2)}(j)\end{cases} \tag{2.18}$$

The maximum of the expected accuracy MEA is given by

$$\begin{aligned}
\mathrm{MEA}_{\alpha} &= \max_{\mathcal{A}} EA_{\alpha^{(a)}}(\mathcal{A}) \\
&= M_{L^{(1)}+1,L^{(2)}+1}
\end{aligned}$$

The corresponding alignment $\mathcal{A} = \mathcal{A}_{\mathrm{MEA}}$ is the MEA solution. Similar to the secondary structure case, the parameter $\alpha^{(a)}$ controls the sensitivity and specificity of the match positions. However, due to the strong constraint on the allowed match positions in Equation2.13, the parameter $\alpha^{(a)}$ is often set to zero so that the sensitivity is maximized.

$p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)})$ can be calculated using the standard pair hidden Markov model (PHMM) of sequence alignment [8]. The model structure of PHMM is shown in Figure 2.1.

$$p^{(a)}(y^{(a)}|x^{(1)}, x^{(2)}) = \frac{p^{(a)}(\tau, x^{(1)}, x^{(2)})}{\sum_{\tau} p^{(a)}(\tau, x^{(1)}, x^{(2)})}$$

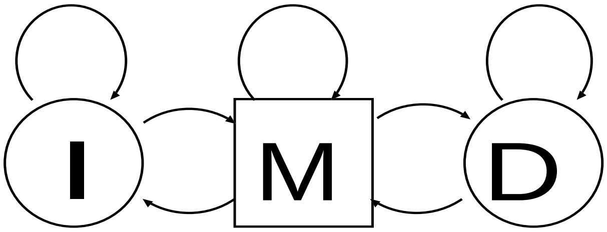


Figure 2.1. The model architecture of PHMM which is used to calculate the match probabilities $p^{(a)}$. The M indicates the match state and I, D indicate the insertion and deletion states, respectively.

$p^{(a)}(\tau, x, y)$ is the joint probability of generating the alignment path τ and given by the product of transition and emission probabilities of the PHMM model. $p^{(a)}(i, j)$ is calculated using the forward and backward algorithms (Algorithm 2. The computation of $p^{(a)}$ requires $\mathcal{O}(L^2)$ time and $\mathcal{O}(L^2)$ memory.

Algorithm 2 Forward and backward algorithms.

Forward:

$$
\begin{aligned}
F_M(i,j) &= e^{s(i,j)}\left[F_M(i-1,j-1) + F_I(i-1,j-1)e^{-d} + F_D(i-1,j-1)e^{-d}\right]\\
F_I(i,j) &= F_M(i,j-1)e^{-d} + F_I(i,j-1)e^{-e}\\
F_D(i,j) &= F_M(i-1,j)e^{-d} + F_D(i-1,j)e^{-e}
\end{aligned}
$$

Backward:

$$
\begin{aligned}
B_M(i,j) &= B_M(i+1,j+1)e^{-s(i+1,j+1)} + B_I(i,j+1)e^{-d} + B_D(i+1,j)e^{-e}\\
B_I(i,j) &= B_M(i+1,j+1)e^{s(i+1,j+1)-d} + B_I(i,j+1)e^{-e}\\
B_D(i,j) &= B_M(i+1,j+1)e^{s(i+1,j+1)-d} + B_D(i+1,j)e^{-e}
\end{aligned}
$$

Chapter 3

Multiple alignment algorithm for structural RNA sequences

3.1. Overview

In this chapter, I propose an efficient algorithm for multiple sequence alignment of structural RNA sequences. The first half of this chapter describes

- the alignment model and the objective function maximized by the Sankoff algorithm.
- an efficient method that considerably reduce the dynamic programming (DP) region to be computed.
- two approximation methods to constrain the DP region
- the multiple alignment procedure such as how the guide tree of progressive alignment are constructed, and the method to align two groups of aligned sequences.
- the probabilistic consistency transformations for match probabilities and base pairing probabilities that improve the accuracy of alignment results.

The second half of this chapter describes the results of that experiments that show the accuracy and speed of the proposed alignment method.

3.2. Background

Because the evolution of a structural RNA gene has the unique characteristic that the substitutions of distant bases are correlated in order to conserve their stem structures, multiple alignment methods should account for such substitution patterns for the accurate detection of conserved structures. The Sankoff algorithm [38] is an alignment algorithm that naturally includes the base pair covariation effect in the alignment model. However, it is not practical to use the Sankoff algorithm in the original form due to its prohibitive computational cost. Hence, there have been intensive studies that investigate practical variations of the Sankoff algorithm in recent years [28],[17],[11],[22],[14],[41],[6]. The algorithms proposed in these studies are roughly categorized into two groups, depending on how the secondary structures are scored in the algorithm.

The algorithms in the first group score the structures using the free energy parameters collected by the Turner group [27]. The algorithms in this group have the advantage of their relatively accurate structure predictions. However, it is difficult for these algorithms to combine the structure energy with the homology information consistently. The pairwise alignment programs Dynalign [28],[41] and Foldalign [14], and the multiple alignment program PMMulti [17] belong to this group.

The second group scores the structures as a part of the probabilistic model called the pair stochastic context free grammar (PSCFG). The advantage of these algorithms is that the parameters that score both the alignments and structures are determined in a unified manner. However, these algorithms have a potential disadvantage that the accuracies of the structure models might be only modest due to the limitations of PSCFG, as compared to those in the first group. The pairwise alignment program Consan [6] and the multiple alignment program Stemloc [22] belong to this group.

These algorithms provides a variety of methods to reduce the huge cost of computation. Dynalign restricts the DP region to a narrow band region so that only similar positions of sequences are compared to each other. Foldalign and PMMulti limit the lengths of subsequences that are compared to each other. Stemloc implements a general method for combining the constraints in the structure space and those in the alignment space that are computed using a Waterman-Eggert style suboptimal alignment algorithm [45]. Consan constrains the DP region by anchoring the points in the DP matrix that have very high posterior probabilities of alignment that are computed by the pair hidden Markov model (PHMM).

However, the computational cost is still quite high even these approximations are applied and It has been impractical to use these programs for aligning sequences having lengths longer than 200 bases. Therefore, several studies have sought for algorithms that circumvent the Sankoff algorithm for fast computation of common secondary structures. For example, the SCARNA program [39] aligns the stem candidate sets extracted from the base pairing probability matrices of two sequences by using very fast dynamic programming algorithm. The RNAcast program predicts common secondary structures from unaligned sequences [36], and the RNAmine algorithm [13] exhaustively enumerates the frequent stem motif patterns from unaligned sequences.

In this chapter, I propose a practical method for aligning multiple RNA sequences based on the Sankoff algorithm. I show that both the alignment quality and the accuracy of the consensus secondary structure prediction from the alignment are the highest among the existing alignment softwares. I also show that our algorithm can align relatively long RNA sequences that have not been computable by other Sankoff-based algorithms. The algorithm is implemented in the software "Murlet."

3.3. Systems and Methods

3.3.1 The Model

I first describe our algorithm for a pairwise sequence alignment. The derivation of our alignment algorithm is guided by two principles.

The first is the principle of extensive preprocessing before applying the Sankoff algorithm. In general, the alignment of structural RNA sequences requires simultaneous consideration of complex information such as base substitution score, gap insertion cost, stacking energy, and various loop energies. If all of these elements are included in the Sankoff model, the computation would be unmanageably slow. Instead, I use the match probability $p^{(a)}$ and the base pairing probability $p^{(b)}$ to score the alignments and structures. Both $p^{(a)}$ and $p^{(b)}$ can be computed by much faster algorithms than the Sankoff algorithm and compactly represent complex information such as sequence homology and structure contexts. This enables us to keep the Sankoff model very simple. Since these quantities $p^{(a)}$ and $p^{(b)}$ do not include the base pair substitution effects, I also apply the base pair substitution matrix $s(i, j, k, l)$ to score the base pair substitution events.

The second principle is the maximal expected accuracy (MEA) principle. Recent studies have shown that the accuracy of the sequence alignment and the secondary structure predictions based on the principle of the maximization of expected accuracy [31] perform better than the predictions made by the conventional maximal likelihood algorithms [4],[35],[24]. A straightforward application of the MEA principle to the Sankoff algorithm would include the calculation of the posterior probabilities of loop match and stem match events by using the inside-outside algorithm. However, such computation is quite demanding because the corresponding Sankoff model would require a large number of states to express the complex homology and structure information, as described in the previous paragraph. Therefore, I instead have adopted a factorized form (Equations 3.1 and 3.2), which is expected to exhibit behavior similar to the posterior probabilities of the Sankoff model.

To give the mathematical definition of our algorithm, I consider the consensus secondary structure annotation $\mathcal{S}$ for each pairwise alignment $\mathcal{A}$ of length L, which consists of sequences x and y of lengths L_x and L_y, respectively.

$$\begin{aligned}
&\mathcal{S} = \mathcal{S}_{\mathcal{A}} = \{\mathcal{L}, \mathcal{P}\} \\
&\mathcal{L} = \{I \in \mathcal{C} | \text{column } I \text{ does not form any base pair}\} \\
&\mathcal{P} = \{(I, J) \in \mathcal{PC} | \text{columns } (I, J) \text{ form a base pair}\}
\end{aligned}$$

where the match columns $\mathcal{C}$ is the set of alignment columns without gap characters, and $\mathcal{PC} = \{(I, J) \in \mathcal{C} \times \mathcal{C} | 1 \leq I < J \leq L\}$ is the set of pairs of match columns $\mathcal{C}$. I consider only the cases where all the base pairs are formed between the match columns. I also ignore pseudo-knotted structures. I assign a score e_L

to each loop column $I \in \mathcal{L}$ and a score e_S to each column pair $(I, J) \in \mathcal{P}$.

$$
\begin{aligned}
e_L(i_I, j_I) =& \gamma_L p^{(a)}(i_I, j_I) q^{(b)}(i_I) q^{(b)}(j_I) && (3.1)\\
e_S(i_I, j_I, i_J, j_J) =& \gamma_S p^{(a)}(i_I, j_I) p^{(a)}(i_J, j_J) \\
& \times p^{(b)}(i_I, i_J) p^{(b)}(j_I, j_J) \\
& \times \exp(s(i_I, j_I, i_J, j_J)) && (3.2)
\end{aligned}
$$

where i_I and j_I represent the sequence positions of sequences x and y aligned at column I. $s(i_I, j_I, i_J, j_J)$ denotes an element of the base pair substitution matrix. γ_L and γ_S are constant coefficients.

For each alignment $\mathcal{A}$ and its consensus structure candidate $\mathcal{S}$, the alignment score $z = z(\mathcal{A}, \mathcal{S})$ is defined as the sum of the loop match scores e_L and the base pair match scores e_S.

$$
z = \sum_{I \in \mathcal{L}} e_L(i_I, j_I) + \sum_{(I,J) \in \mathcal{P}} e_S(i_I, j_I, i_J, j_J)
$$

The alignment result $(\mathcal{A}_{\max}, \mathcal{S}_{\max})$ is obtained by taking the maximum of the score $z = z_{\max}$ among all the alignments and structures.

To compute the maximum of $z(\mathcal{A}, \mathcal{S})$, I have adopted the following DP of the Sankoff algorithm.

$$
M_{i,j,k,l} = \max \begin{cases}
M_{i+1,j+1,k-1,l-1} + e_S(i, j, k, l) \\
M_{i+1,j+1,k,l} + e_L(i, j) \\
M_{i,j,k-1,l-1} + e_L(k, l) \\
M_{i+1,j,k,l} \\
M_{i,j+1,k,l} \\
M_{i,j,k-1,l} \\
M_{i,j,k,l-1} \\
M_{i,j,u,v} + M_{u+1,v+1,k,l} \text{ for } i < u < k,\ j < v < l
\end{cases} \qquad (3.3)
$$

After the DP computation, the maximum of the score is obtained by $z_{\max} = M_{1,1,L_x,L_y}$. The computation of Equation 3.3 requires $\mathcal{O}(L^6)$ time and $\mathcal{O}(L^4)$ memory.

Note that the alignment result is defined in terms of the score $z(\mathcal{A}, \mathcal{S})$ and is independent of the details of the grammar of the Sankoff algorithm, when the algorithm of Equation 3.3 is interpreted as the Cocke-Younger-Kasami (CYK) algorithm of PSCFG. I can use an arbitrary grammar to compute the alignment, provided that the grammar can parse all the alignments and structures, and that it does not modify the score system. The latter condition implies that the model cannot have any transition scores and that the left and right emission scores have to be exactly the same. The independence of the alignment result on a particular grammar also indicates that there are no problems with the ambiguity of the grammar. For an ambiguous grammar, two or more parse trees correspond to the same alignment and structure. Since the score only depends on the alignment and structure, which of the parse trees is chosen depends on the detailed order of computations. This indicates that the obtained parse tree has little meaning. However, the alignment and its associated structure are unique, and they are sufficient for our purpose.

In contrast, the computations of the match and pair probabilities are affected by the redundant enumeration of the same alignment and structure. However, both the forward-backward algorithm of the model of Figure 2.1 and the McCaskill algorithm enumerate all the alignments and structures without redundancies. Therefore, the whole algorithm is free from the redundancy problems.

3.3.2 Reduction of DP Region

Because the loop match score e_L and the base pair match score e_S are both proportional to the match probability $p^{(a)}$, I restrict the $L_x \times L_y$ DP region to a smaller one that includes all the positions with $p^{(a)}(k,l) > \epsilon$, where ϵ is a prespecified threshold value.

For each alignment $\mathcal{A}$, let $\mathcal{M}^{\epsilon}_{\mathcal{A}}$ denote the set of match positions in the alignment $\mathcal{A}$ that satisfy $p^{(a)}(i,j) > \epsilon$. For a given initial alignment path and a threshold value $\epsilon > 0$, I then define the restricted DP region as the smallest region in the DP matrix that satisfies the following conditions.

1. The region is simply connected, that is, the region has no holes.

2. The region includes the initial alignment path.

3. For each alignment path $\mathcal{A}$ with $\mathcal{M}^{\epsilon}_{\mathcal{A}} \neq \emptyset$ in the full DP region, there exists an alignment $\mathcal{A}'$ in the restricted DP region that satisfies $\mathcal{M}^{\epsilon}_{\mathcal{A}'} = \mathcal{M}^{\epsilon}_{\mathcal{A}}$.

I have described the algorithm for computing the restricted DP region in the supplementary information. The third condition implies that if all the match probabilities $p^{(a)}(i,j)$ that are not greater than ϵ are set to zero, then there always exists an alignment in the restricted DP region that has the same score as the optimal score of the Sankoff algorithm in the full DP region. It implies that for a sufficiently low threshold value ϵ (I use $\epsilon = 0.0001$ throughout the study), the restriction of the DP region rarely cause missing the optimal alignment.

If two sequences are highly similar, the match probabilities concentrate along a specific diagonal in the DP matrix and the reduction of the DP region is quite significant. As shown in the later section, the elapsed time and memory are drastically reduced for similar sequences.

Previous studies have also considered to restrict the DP region using PHMM [6],[22]. In particular, our reduction method is a particular case of a more general method proposed by Holms *et al.* [22]. However, our method is different from their algorithms in two respects. The first is that our method is more conservative than their algorithms, since all the likely positions are kept in the DP region rather than only highly possible regions are selected as in their methods. The second point is that the score system of the Sankoff algorithm is more closely tied to the PHMM that is used to reduce the DP region. Since the loop match score e_L and the pair match score e_S are both proportional to the match score $p^{(a)}$, there are no contribution to the total score from the positions with zero match probabilities $p^{(a)}(i,j) = 0$. On the other hand, in their algorithm, the score system of the Sankoff algorithm and the score system of PHMM is not directly related. Hence, it is possible that the total alignment score have a large contribution from the positions with vanishing match probabilities. For these reasons, our restriction method is expected to have less possibility of missing the optimal alignment than their algorithms.

3.3.3 Approximation Methods

Most of the alignment softwares based on the Sankoff algorithm provide optional parameters to approximate the DP and to control the trade-offs between the computational cost and the alignment accuracy [17],[11],[28],[22],[14]. Murlet provides two original approximations that constrain the DP region: the strip and skip approximations.

For a given initial alignment path, *the strip approximation* constrains the DP region to a strip region of fixed width δ around the alignment path. If the strip width δ is equal to one, then the resulting alignment after the DP computation is the same as the initial alignment, as in the QRNA software [37]. If a diagonal path is specified as the initial alignment path, then the strip approximation corresponds to the band alignment that calculates only the region $|k - l| < \delta$ for row k and column l in the DP matrix.

The previous version of Dynalign [28] software implements the band approximation. The limitation of the band approximation is that the band width cannot be smaller than the difference $|L_x - L_y|$ of two sequences. The approximation method that is adopted by Foldalign and PMMulti also suffers from the similar limitation. The recent version of Dynalign [41] has modified the definition of the band region as $|k(L_y/L_x) - l| < \delta$ so that the band width can take values as small as one. The strip approximation is more general as compared to these approximations, because the initial path can be arbitrarily far from the main diagonal of the DP matrix and the strip width can be set to one irrespective of the difference of sequence lengths.

If the restriction of the DP region by match probabilities is not applied, the computational cost of the Sankoff algorithm is reduced by $(\delta/L)^3$ times in time and $(\delta/L)^2$ times in memory.

The skip approximation constrains the points of the bifurcation calculations (the last line of Equation 3.3) to a restricted set of positions in the DP region.

$$\begin{aligned} &M_{i,j,u,v} + M_{u+1,v+1,k,l} \text{ for } i < u < k,\ j < v < l \\ &\Longrightarrow \text{ if } (i,j),(k,l) \in \mathcal{K},\ M_{i,j,u,v} + M_{u+1,v+1,k,l} \text{ for } (u,v) \in \mathcal{K} \end{aligned} \tag{3.4}$$

That is, the bifurcation calculation is performed only when the end positions (i, j) and (k, l) are in the skip set $\mathcal{K}$, and the only case considered is the one where the mid position (u, v) is in the skip set $\mathcal{K}$. The skip set $\mathcal{K}$ is a set of grid positions in the DP region that is defined as follows.

$$\mathcal{K} = \{(i,j) | i \equiv 1 \pmod{\kappa}, j \equiv \tau(i) \pmod{\kappa}\}$$

where $\tau(i)$ is a point on the initial alignment path at row i, and $\kappa > 0$ is a given parameter. $\kappa = 1$ corresponds to the full DP in the DP region, and in the limit $\kappa \to \infty$, the algorithm can only parse non-bifurcating stem structures just as the earlier version of the Foldalign software [11]. The bifurcation part of computation, which requires $\mathcal{O}(L^6)$ time and $\mathcal{O}(L^4)$ memory, decreases by $1/\kappa^6$ times in time and $1/\kappa^4$ times in memory with the skip approximation. If the skip size κ is three or more, the bifurcation part is not a dominant factor of computation for aligning sequences of lengths shorter than 500 bases. In such cases, the leading contribution to the total memory comes from the $\mathcal{O}(L^4)$ memory that is required to store the traceback pointers. In the Murlet implementation, only one byte is required to store the traceback information for each DP iteration. Note that only the $\mathcal{O}(L^3)$ memory is required in order to calculate the first seven lines of Equation 3.3. However, this part of computation requires $\mathcal{O}(L^4)$ time and dominates the total computation time.

The skip approximation is considered because the occurrence frequency of bifurcations in the parse tree is small as compared to the lengths of the RNA sequences, even though the bifurcation calculation is the most compute-intensive part of the Sankoff algorithm. However, the skip approximation may miss a few base pairs if two neighboring stems are close to each other and no skip points are placed between them.

For a given strip width δ and skip size κ, the DP region of the Sankoff algorithm is determined as follows (see Figure 3.1). First, the initial alignment path is determined (Figure 3.1(a)) by the following DP algorithm which is an application of the MEA principle to the PHMM.

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + p^{(a)}(i,j) \\ M_{i-1,j} \\ M_{i,j-1} \end{cases}$$

I refer to the alignment obtained by this computation as the PHMM-MEA alignment. Next, the DP region is constrained to the strip region around the initial alignment path (Figure 3.1(b)). The DP region is further constrained by stripping away the side regions with low match probabilities $p^{(a)}$ (Figure 3.1(c)). Finally, the skip set $\mathcal{K}$ is determined within the DP region using the initial alignment

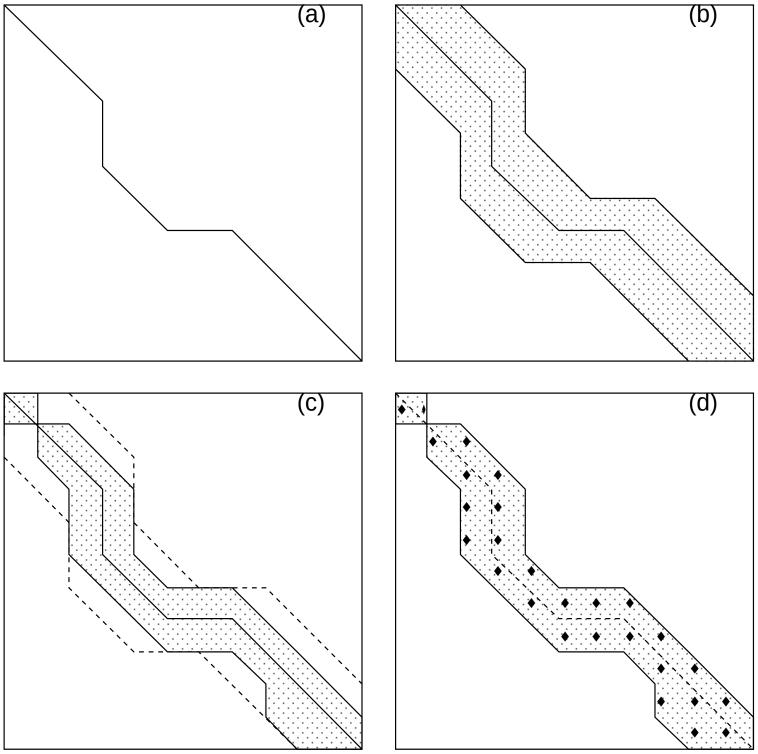


Figure 3.1. Procedure to constrain the DP region of the Sankoff algorithm. (a) The initial DP alignment is calculated by the PHMM-MEA method. (b) The DP region is constrained to a strip region around the initial DP path. (c) The DP region is reduced further by stripping away the regions with low match probabilities. (d) The skip set is fixed inside the DP region.

path (Figure 3.1(d)).

It is tedious to determine the appropriate strip width δ and skip size κ for each sequence pair being aligned. Murlet estimates the allocated memory and the computational time for each pairwise alignment and automatically determines the strip width and skip size so that the DP region is maximal under the given memory and time limits specified by the user.

The computation time t is estimated by the following formula.

$$t = a\mu_{\text{traceback}} + b\mu_{\text{bifurcation}}^{\frac{6}{4}} \tag{3.5}$$

where $\mu_{\text{traceback}}$ is the size of the $\mathcal{O}(L^4)$ memory which is needed to store traceback information of the Sankoff algorithm, and $\mu_{\text{bifurcation}}$ is the $\mathcal{O}(L^4)$ memory require to store the scores of the child states of the bifurcation. a and b are fitting parameters. $\mu_{\text{bifurcation}}^{\frac{6}{4}}$ is the estimated number of bifurcation calculations (Equation 3.4).

Figure 3.2 shows a scatter plot of the estimated time (x-axis) and the real time (y-axis). I used the pairwise alignments derived from the dataset of Table 3.1. I varied the strip width δ from 0.1 to 0.5 and skip size κ from 1 to 5, and measured the elapsed time for the computation of pairwise alignments. As seen in the figure, the computation time can be estimated with a reasonable accuracy.

3.3.4 Probabilistic Consistency Transformations

For three or more sequences in the same sequence family, Do *et al.* introduced the probabilistic consistency transformation (PCT) of match probability matrices [3], which is defined by the formula.

$$p_{x,y}^{(a)\text{new}}(i,j) \leftarrow \frac{1}{N} \sum_{w \in X, k} p_{x,w}^{(a)\text{old}}(i,k) p_{w,y}^{(a)\text{old}}(k,j)$$

where x, y, w represent sequences in X, and i, j, k are the sequence positions in sequences x, y, w. $p^{(a)\text{old}}$ and $p^{(a)\text{new}}$ are the match probabilities before and after the transformation, respectively. This computation requires $\mathcal{O}(N^3L^3)$ time for N sequences of length L. By this transformation, the match probabilities $p_{x,y}^{(a)}(i,j)$ is increased if there are positions in other sequences that are likely to match with i

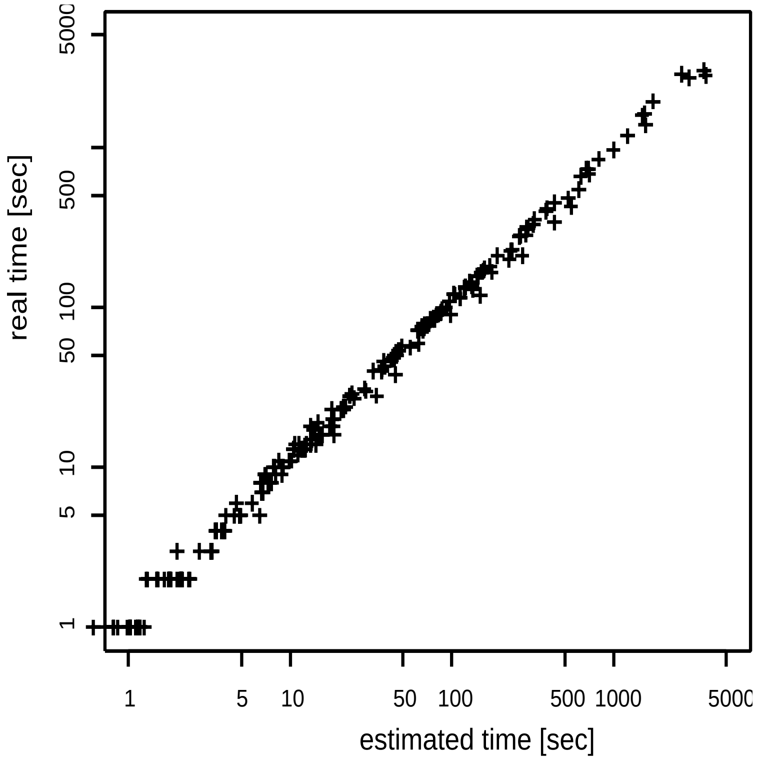


Figure 3.2. A scatter plot showing the accuracy of the estimation of computation time. The x-axis is the estimated time in seconds, computed by Equation 3.5. The y-axis is the elapsed time in seconds for the pairwise alignment. The number of the data points is 246.

and j, and is decreased if there are no such positions. Hence, the transformation adds the family-specific homology information to the match probabilities.

Here, I propose the PCT for the base pairing probability matrices, which is defined by the formula.

$$p_x^{(b)\text{new}}(i,j) \leftarrow \frac{1}{N} \sum_{w \in X,k,l} p_{x,w}^{(a)}(i,k) p_{x,w}^{(a)}(j,l) p_w^{(b)\text{old}}(k,l)$$

The computation requires $\mathcal{O}(N^2L^4)$ time. The new loop probabilities $q_x^{(b)\text{new}}(i)$ are computed by applying Equation 2.6 to $p_x^{(b)\text{new}}(i,j)$.

$$\begin{aligned} q_x^{(b)\text{new}}(i) &= 1 - \frac{1}{N} \sum_{w \in X} \left[\sum_{1 \le j < i} t_{xw}(j,i) + \sum_{i < j \le L_x} t_{xw}(i,j) \right] \\ t_{xw}(i,j) &= \sum_{1 \le k < l \le L_w} p_{x,w}^{(a)}(i,k) p_{x,w}^{(a)}(j,l) p_w^{(b)}(k,l) \end{aligned}$$

Then, $q_x^{(b)\text{new}}(i)$ assumes a value between zero and one.

$$0 \le q_x^{(b)\text{new}}(i) \le 1 \tag{3.6}$$

This formula justifies in treating the transformed matrices $p_x^{(b)\text{new}}(i,j)$ as the pair probability matrices.

As in the case of match probabilities, the transformation adds the family-specific structure information to the base pairing probabilities. I show in the later section, the PCT for the match probabilities considerably improve the alignment accuracy.

The formula 3.6 can be proved as follows. Since $t_{xw}(i,j) \ge 0$, the inequality $q_x^{(b)\text{new}}(i) \le 1$ is obviously satisfied. Hence, I prove only the inequality,

$$\sum_{1 \le j < i} t_{xw}(j,i) + \sum_{i < j \le L_x} t_{xw}(i,j) \le 1 \tag{3.7}$$

for fixed i. The first term of the above formula can be bounded from above as follows:

$$\begin{aligned} \sum_{1 \le j < i} t_{xw}(j,i) &= \sum_{1 \le k < l \le L_w} \left[\sum_{1 \le j < i} p_{x,w}^{(a)}(j,k) \right] p_{x,w}^{(a)}(i,l) p_w^{(b)}(k,l) \\ &\le \sum_{1 \le k < l \le L_w} p_{x,w}^{(a)}(i,l) p_w^{(b)}(k,l) \end{aligned}$$

The expression in the square bracket is not greater than one since it is the probability that the position k of sequence w is aligned to the range between 1 and $i-1$ of sequence x. Similarly, the second term satisfies the inequality.

$$\sum_{i<j\leq L_x} t_{xw}(i,j) \leq \sum_{1\leq k<l\leq L_w} p^{(a)}_{x,w}(i,k)p^{(b)}_{w}(k,l)$$

Hence, the left-hand-side lhs of Equation 3.7 satisfies the formula:

$$\begin{aligned} lhs &\leq \sum_{1\leq k\leq L_w} p^{(a)}_{x,w}(i,k)\left[\sum_{1\leq l<k} p^{(b)}_{w}(l,k) + \sum_{k<l\leq L_w} p^{(b)}_{w}(k,l)\right] \\ &\leq \sum_{1\leq k\leq L_w} p^{(a)}_{x,w}(i,k) \\ &\leq 1 \end{aligned}$$

In the above formula, the expression inside the square bracket is not greater than one because it is the probability that the position k forms any base pair with other positions. Further, since the right-hand-side of the second inequality represents the probability that the position i of sequence x is aligned to any position of sequence w, the last inequality follows. Thus, the formula 3.6 is proved.

The PCTs for $p^{(a)}$ and $p^{(b)}$ are performed for the sparse matrix representations of the probability matrices to reduce the computation time.

3.3.5 Multiple Alignment Procedure

I now describe the multiple alignment procedure. Let N be the number of input sequences X. First, the base pairing probability matrices and the match probability matrices are computed for each sequence and each pair of sequences, respectively. Next, PCT for the match probabilities is performed, which is followed by the PCT for the pair probabilities using the transformed match probabilities. For each pair of sequences, the similarity between them is defined as the score of the Sankoff algorithm along the PHMM-MEA alignment path. Using this similarity measure, the guide tree is constructed by using the unweighted pair group method (UPGMA) clustering algorithm. The progressive alignment is then performed using the guide tree. To compute two groups of aligned sequences, the base pairing probabilities are averaged over all the sequences of each group. Further, the match probabilities are averaged over all the pairs of sequences between the two groups. The base pair substitution score $s(i_I, j_I, i_J, j_J)$ in Equation 3.2 is computed as the sum of the corresponding values for all the pairs of sequences between the groups. The proportionality constants γ_L and γ_S in Equations 3.1 and 3.2 are set as

$$\gamma_L = 0.005$$
$$\gamma_S = 4.0 N_1 N_2$$

where N_1 and N_2 are the number of sequences in the two groups.

For the computation of the match probabilities, I used the ProbCons software (version 1.10) [3]. For the computation of the base pairing probabilities, I used the RNAAlifold program of the Vienna RNA package (version 1.5) [19],[16]. The base pair substitution matrix is extracted from the Stemloc software in the DART package [22].

3.3.6 The Dataset

I collected the test dataset from the Rfam database [12]. I used only the hand-curated seed alignments with the consensus structures published in literatures. For each sequence family, I generated up to 1000 random combinations of 10 sequences. I then removed the alignments with mean pairwise sequence identity more than 95 %. Because I are considering the global multiple alignment problem, I removed the alignments that contain gap characters more than 30 % of total alignment characters. I also removed the alignments with gap characters less than 5 % of the total alignment characters, since for such alignments, the algorithms that merely penalize or forbid the gap insertions show high accuracies. I found it difficult to collect completely exclusive alignment set for many sequence families. Therefore, I removed only the alignments that share more than 30 % of sequences with another alignment. Inspecting the number of families and the number of sub-alignments available for each family, I chose the dataset shown in Table 3.1.

The dataset consists of 85 multiple alignments of 10 sequences. The number of sequence families is 17 and for each sequence family, there are five multiple alignments. The dataset is reasonably diverged and its mean length varies from

54 bases to 291 bases and mean pairwise sequence identities varies from 40 % to 94 %.

Previously, Gardner et al. [10] have collected a dataset of multiple alignments of structural RNA sequences in order to benchmark various alignment algorithms. However, their dataset is not appropriate for our purpose for two reasons. The first reason is that their dataset does not have sufficient variety in sequence lengths, since the dataset consists of short ($<$ 120 bases) sequence families (Group II introns, 5S rRNA, tRNA, and U5 splisomal RNA) and a relatively long ($\approx$ 300 bases) family (SRP). The second reason is that the distribution of sequence identity is strongly biased, and in particular, few alignments have low ($< 55\%$) sequence identities. As they described at their web site (http://www.binf.ku.dk/ pgardner/bralibase/), this is caused by the erroneous calculation of sequence identities. Because of the lack of diversity in both the sequence length and the sequence identity, I have used only our original dataset for the evaluation.

3.3.7 Accuracy Measure

The accuracy of the alignments is measured by the standard sum-of-pairs score (SPS) [1]. To measure the efficiency of the structural alignment, the consensus structures for the alignment results are predicted using the Pfold program ([24]). The Matthews correlation coefficients (MCC) are then calculated for the predictions ([29]). MCC is defined by the formula

$$\mathrm{MCC} = \frac{\mathrm{tp} \cdot \mathrm{tn} - \mathrm{fp} \cdot \mathrm{fn}}{\sqrt{(\mathrm{tp}+\mathrm{fp})(\mathrm{tp}+\mathrm{fn})(\mathrm{tn}+\mathrm{fp})(\mathrm{tn}+\mathrm{fn})}}$$

where tp is the number of correctly predicted base pairs, tn is the number of base pairs that are correctly predicted as unpaired, fp is the number of incorrectly predicted base pairs, and fn is the number of true base pairs that are not predicted. Note that tn is computed in units of base pairs and is very large in most cases. The numbers are computed by assigning both reference and predicted consensus structures to each sequence using the alignment and then counting the matches and mismatches of base pairs for all the sequences.

Since the computation of MCC uses the external program Pfold, the results may be skewed by the compatibility of the programs with the Pfold software.

Therefore, I also measured the efficiency of structural alignment with the novel indicators SSS, SQS and PCS that quantify how well the true stems are aligned to each other. These indicators do not depend on the structure predictions to the alignment results and only use the reference and subject alignments and the structures annotated to the reference alignments. Therefore, these measures do not depend on any particular external program or adjustable parameters. They are defined analogously to SPS and the column score (or TC score) [40],[1], which are frequently used for the evaluation of the sequence alignments.

The sum-of-quadruples score (SQS) is defined as the fraction of the count of the pairs of *base pairs* that are correctly aligned as in the reference alignment. The counts are computed for all the pairs of sequences. The base pairing positions of each sequence are derived from the annotated consensus secondary structure to the reference alignment in the obvious manner. The sum-of-stem-pairs score (SSS) is defined similarly, but the criterion of a count is looser and allows the match of base pairs at different alignment columns in the reference alignment. In other words, it counts one if a base pair is aligned to another base pair, irrespective of their alignment columns in the reference alignment. The pair column score (PCS) is the fraction of the number of pair columns, for each of which there exists an identical pair columns in the reference alignment that form a base pair. SQS and PCS take values between zero and one, and they are equal to one if the subject alignment is identical to the reference alignment. SSS is also a non-negative number and is equal to one if the alignment is identical to the reference alignment. It is also equal to or less than one if all the stem regions in the reference alignment do not contain gap characters. However, it might be larger than one when two or more sequences have gap characters in the stem regions of the reference alignment. The mathematical definitions and examples of computations for these measures are presented in the supplementary information.

3.4. Results

3.4.1 Comparison of Programs

Table 3.1 shows a comparison of the accuracy of the alignment for various alignment algorithms. The first three columns in the table show the Rfam family name, mean sequence length, and mean pairwise percent identity. The remaining the columns show the SPS and MCC values for various algorithms: ProbCons [3] and ClustalW [40] are the alignment softwares based on PHMM. Murlet, Stemloc [22], and PMMulti [17] are based on the Sankoff algorithm. I set the time limit for each pairwise alignment to 10 min. The other softwares are used with the default option. The computations are performed on a Linux machine equipped with dual AMD Opteron 850 2.4 GHz processors and 4 GB RAM. Due to the formidable time and memory consumption of Stemloc and PMMulti for longer sequence families, I limit the time and the maximal resident physical memory of the process to 500 min and 3.5 GB, respectively. I terminated the computation if the process exceeded the time or memory limit. If some of the five alignments in the family are not returned within the limits, the fraction of the alignments returned is indicated inside round brackets.

The last four rows indicate the average values of SPS and MCC for each software. "Average (all)" indicates the average values taken over all the families. "Average (Stemloc)," "Average (PMMulti)," and "Average (common)" represent the average values taken over the partial alignment set for which Stemloc, PMMulti, and both returned results, respectively. The fractions of the number of alignments to the whole dataset are indicated in round brackets.

Table 3.1 shows that among the softwares examined, the performance of Murlet was the best in terms of both the alignment accuracy (SPS) and the accuracy of the structure prediction (MCC). Although the SPS values of ProbCons and the MCC values of Stemloc are relatively close to those of Murlet, the MCC values of ProbCons and the SPS values of Stemloc are much lower than the corresponding values of Murlet. The table also shows that the accuracies of ClustalW and PMMulti are lower than those of the other programs. Stemloc and PMMulti could not align most of the RNA sequences having lengths longer than 150 bases within the time and the memory limit. In almost all the cases, the failures of Stemloc and PMMulti are caused by excessive memory requirements. The fact that the SPS values of Murlet are higher than those of ProbCons indicates that the inclusion of stem conservation in the alignment model really does improve the alignment quality.

Table 3.1. Comparison of the SPS and MCC values for several multiple alignment programs. The first three columns list the Rfam family name, the mean sequence length of each family, and the mean pairwise percentage identity. The remaining columns show the SPS and MCC values of the alignment results. For each row, the highest values of SPS and MCC are shown in bold type face.

Family name	Length	% identity	Murlet	ProbCons	ClustalW	Stemloc	PMMulti
			SPS / MCC	SPS / MCC	SPS / MCC	SPS / MCC	SPS / MCC
UnaL2	54	73	**0.97** / 0.44	**0.97** / **0.46**	0.85 / 0.36	0.92 / 0.41	0.70 / 0.24
SECIS	64	41	**0.73** / **0.74**	0.68 / 0.58	0.35 / 0.00	0.64 / 0.69	0.35 / 0.53
tRNA	73	45	0.91 / 0.95	0.88 / 0.91	0.63 / 0.66	**0.92** / **0.97**	0.70 / 0.82
sno_14q_I_II	75	64	**0.93** / **0.79**	**0.93** / 0.75	0.83 / 0.52	0.83 / 0.74	0.47 / 0.61
SRP_bact	93	47	**0.62** / 0.64	**0.62** / 0.59	**0.62** / 0.60	0.53 / **0.70** (3/5)	0.46 / 0.56 (4/5)
THI	105	55	**0.84** / **0.76**	**0.84** / 0.73	0.59 / 0.46	0.79 / 0.75	0.59 / 0.54
S_box	107	66	**0.90** / **0.84**	**0.90** / 0.83	0.83 / 0.79	0.85 / **0.84**	0.51 / 0.62
5S_rRNA	116	57	0.87 / **0.61**	0.87 / 0.59	0.83 / 0.54	**0.88** / **0.61** (4/5)	0.58 / 0.50
Retroviral_psi	117	92	**0.98** / 0.88	**0.98** / 0.88	0.97 / 0.88	0.96 / **0.89**	0.87 / 0.76
RFN	140	66	**0.92** / 0.67	0.91 / **0.68**	0.84 / 0.65	0.88 / 0.67	0.62 / 0.56 (3/5)
5_8S_rRNA	154	61	**0.90** / **0.36**	0.89 / 0.29	0.80 / 0.14	0.75 / 0.24 (1/5)	0.69 / 0.23 (3/5)
U1	157	59	**0.79** / **0.71**	0.78 / 0.65	0.73 / 0.60	- (0/5)	0.55 / 0.53 (2/5)
Lysine	181	49	**0.78** / **0.85**	0.76 / 0.77	0.61 / 0.51	- (0/5)	0.41 / 0.58 (3/5)
U2	182	62	**0.76** / **0.78**	**0.76** / 0.63	0.67 / 0.47	0.60 / 0.49 (1/5)	- (0/5)
T-box	244	45	**0.51** / **0.65**	**0.51** / 0.59	0.35 / 0.36	- (0/5)	- (0/5)
IRES_HCV	261	94	**0.98** / **0.74**	**0.98** / **0.74**	**0.98** / **0.74**	- (0/5)	- (0/5)
SRP_euk_arch	291	40	**0.44** / **0.60**	0.41 / 0.35	0.35 / 0.25	- (0/5)	- (0/5)
Average (all) (85/85)			**0.81** / **0.71**	0.80 / 0.65	0.70 / 0.50	-	-
Average (Stemloc) (49/85)			**0.86** / **0.71**	0.85 / 0.66	0.73 / 0.50	0.79 / 0.67	-
Average (PMMulti) (50/85)			**0.86** / **0.71**	0.85 / 0.67	0.73 / 0.51	-	0.58 / 0.54
Average (common) (46/85)			**0.87** / **0.69**	0.86 / 0.66	0.74 / 0.50	0.81 / 0.68	0.59 / 0.54

Table 3.2 shows a comparison of the SSS, SQS, and PCS for different softwares. The test sets are the same as those in the last four rows of Table 3.1. The superiority of Murlet compared to the other programs is more obvious for these measures. Moreover, Murlet is the only Sankoff-based program that performs better than the PHMM-based ProbCons software. We have also performed the one-sided Fisher's sign test to see the statical significance of the superiority of Murlet as compared to other programs. Table 3.3 shows the p values of the Fisher's sign test that tests the null hypothesis that Murlet is not more accurate than each of the other programs. The first column shows the name of the program compared. The other columns show the p values that measure the unlikeliness of the null hypothesis. The number of observations that the accuracy of Murlet is better (+) or worse (-) than that of the program being compared is also shown in the round brackets. Table 3.3 shows that the accuracies of Murlet are significantly better than the other programs in almost all the cases. Only the MCC and SSS values of Murlet are less significant as compared to those of Stemloc, partly due to the small size of the data that Stemloc has returned any alignment results.

Tables 3.1, 3.3, and 3.2 indicates that for the structural alignment of RNA sequences, Murlet is the best among the examined programs.

3.4.2 Reduction of Time and Memory

Figure 3.3 shows the memory and time consumption of the three Sankoff-based algorithms. Each data point corresponds to a sequence family shown in Table 3.1. The x-axis represents the mean sequence length of the sequence family, and the y-axis represents the maximal resident physical memory in mega bytes (left) and the elapsed time in minutes (right). The figure shows that the memory consumption of Stemloc and PMMulti drastically increases for sequences above 100 bases in length, and these programs cannot align sequences above 200 nucleotides within the limits of 3.5 GB and 500 min. In contrast, Murlet can align 10 sequences of the SRP_euk_arch family of mean length 291, within a realistic memory (570 MB) and time. (32 min).

Figure 3.4 shows the dependence of the reduction of time and memory requirements on the sequence identities. I used 188 multiple alignments of four sequences collected from the Hammerhead_3 ribozyme family in the Rfam database. I com-

Table 3.2. Comparison of the accuracy of structural alignments using the proposed accuracy measures. The test sets are the same as those shown in the last four rows of Table 3.1. For each alignment set and accuracy measure, the highest value of the measure is shown in bold type face.

	Program	SSS	SQS	PCS
Average (all)	Murlet	**0.81**	**0.79**	**0.55**
	ProbCons	0.75	0.75	0.52
	ClustalW	0.60	0.60	0.34
Average (Stemloc)	Murlet	**0.84**	**0.83**	**0.59**
	ProbCons	0.79	0.79	0.56
	ClustalW	0.63	0.63	0.39
	Stemloc	0.78	0.76	0.50
Average (PMMulti)	Murlet	**0.84**	**0.83**	**0.59**
	ProbCons	0.80	0.80	0.56
	ClustalW	0.63	0.63	0.36
	PMMulti	0.58	0.51	0.22
Average (common)	Murlet	**0.85**	**0.84**	**0.60**
	ProbCons	0.81	0.81	0.58
	ClustalW	0.65	0.65	0.39
	Stemloc	0.81	0.79	0.53
	PMMulti	0.58	0.52	0.23

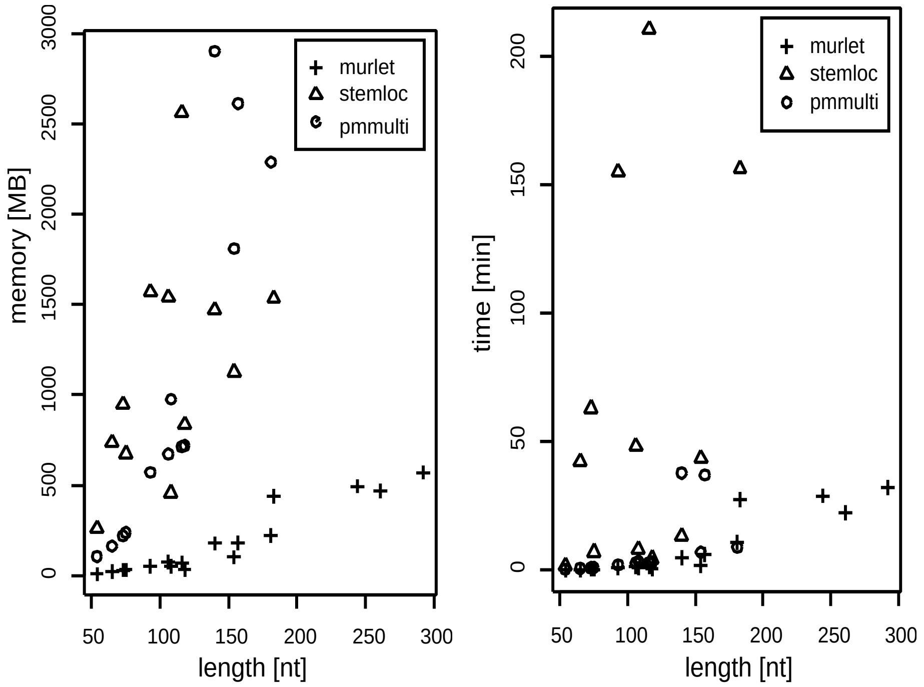


Figure 3.3. Elapsed time and the maximal resident memory for computing alignments of Table 3.1. In both figures, x-axis is the mean length of the sequence families. y-axes are the maximal resident physical memory of the process in mega bytes (left) and the elapsed time in minutes. Each data point represents the specific sequence family of Table 3.1. Only the alignments returned correctly are plotted.

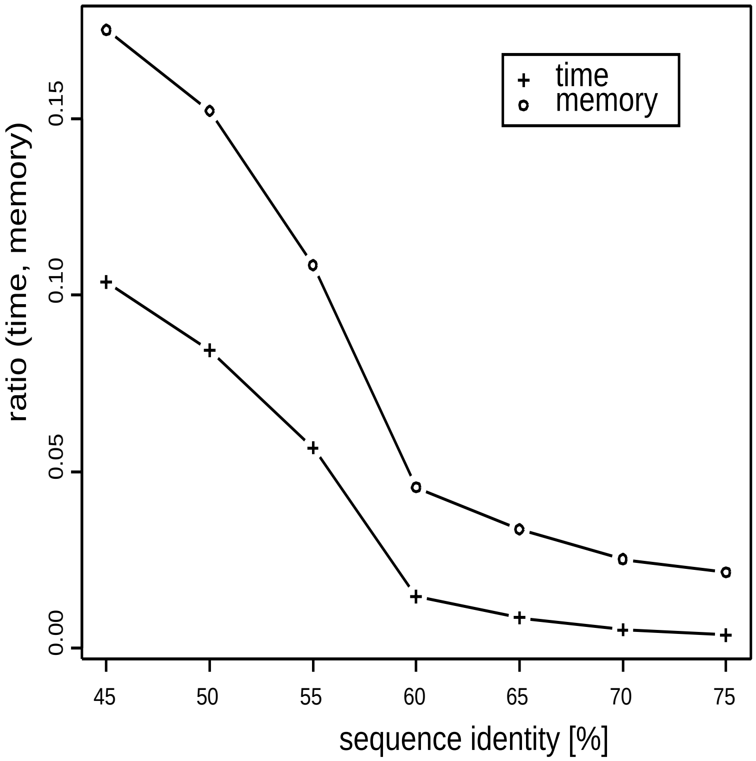


Figure 3.4. Dependence of the reduction of time and memory on the sequence identity. The dataset contains 188 multiple alignments of four sequences collected from the Hammerhead_3 ribozyme family in the Rfam database. Their mean length is 55 bases. The x-axis represents the mean pairwise sequence identity and the y-axis represents the ratio of the estimated time and allocated memory for the DP calculation between the full DP and the DP in the reduced DP region. The data points are categorized into bins of width 5 %, and the mean values of the bins are plotted.

pared the estimated time and the allocated memory between the full DP region and the region reduced by the match probabilities. For all 188 alignments, the two cases returned exactly the same alignment results. The mean SPS and MCC values were 0.87 and 0.85, respectively. The ratios of time and memory were binned for each five percent segment of the sequence identity, and the mean value for each bin was plotted. The figure shows that for sequence identities larger than 60 %, the time and memory requirements are hundreds of times smaller than those in the full DP case. On the other hand, for sequence identities less than 60 %, the required time and memory increase with the decrease of sequence identities, though they are still an order of magnitude smaller than those of the full DP case.

3.4.3 Effects of Probabilistic Consistency Transformations

Figure 3.5 shows density plots of the match probability distribution. The probabilities of the left figure are computed using the forward-backward algorithm of PHMM. The sequences are taken from the tRNA family shown in Table 3.1. The figure on the right represents the probabilities after the consistency transformation. Although the dense regions are broadened by the transformation, they are still concentrated around the main diagonal of the DP matrix.

Figure 3.6 shows an example of the true secondary structure of tRNA (left) and the corresponding base pairing probability matrices (right). The base pairing probability matrix computed by the McCaskill algorithm is shown in the lower-left part of the figure on the right and that obtained after PCT is shown in the upper-right part of the matrix. As indicated by the arrow in the figure, the McCaskill algorithm fails to identify one of the four stems of tRNA. PCT remedies this failure by adding small probabilities to this region.

Table 3.4 shows the effects of the PCTs for the probabilities $p^{(a)}$ and $p^{(b)}$ on the alignment accuracies. For all the measures, the accuracies are the highest when PCT is applied to both the match and pair probabilities. The figure also shows that the PCT for the pair probabilities are more significant than the PCT for the match probabilities, and the latter is only effective when the former is also performed. This indicates that incorrect base pairs are frequently predicted by the McCaskill algorithm and considerably degrade the quality of alignment.

Table 3.3. The p values of the Fisher's sign test that tests the null hypothesis that Murlet is not more accurate than each of the other programs. The first column shows the name of the program compared. The other columns show the p values that measure the unlikeliness of the null hypothesis. The number of observations that the accuracy of Murlet is better (+) or worse (-) than that of the program being compared is also shown in the round brackets.

Program	SPS	MCC	SSS	SQS	PCS
	p-value(+/−)	p-value(+/−)	p-value(+/−)	p-value(+/−)	p-value(+/−)
ProbCons	0.0033 (52/27)	5e-7 (58/16)	8e-13 (61/6)	9e-11 (58/8)	0.022 (28/14)
ClustalW	5e-15 (75/8)	2e-12 (70/10)	2e-16 (75/6)	5e-15 (72/7)	3e-10 (58/9)
Stemloc	3e-8 (43/6)	0.11 (26/17)	0.11 (25/16)	0.0083 (28/12)	0.017 (26/12)
PMMulti	3e-17 (55/0)	0.0092 (51/29)	6e-17 (54/0)	6e-17 (54/0)	9e-16 (50/0)

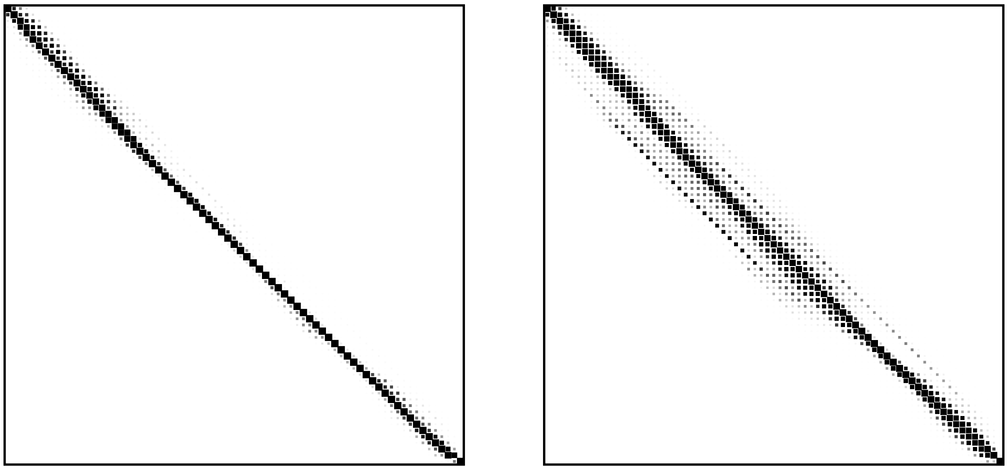


Figure 3.5. PCT for match probabilities. The figures on the left and right are the match probabilities before and after the probabilistic consistency transformation, respectively.

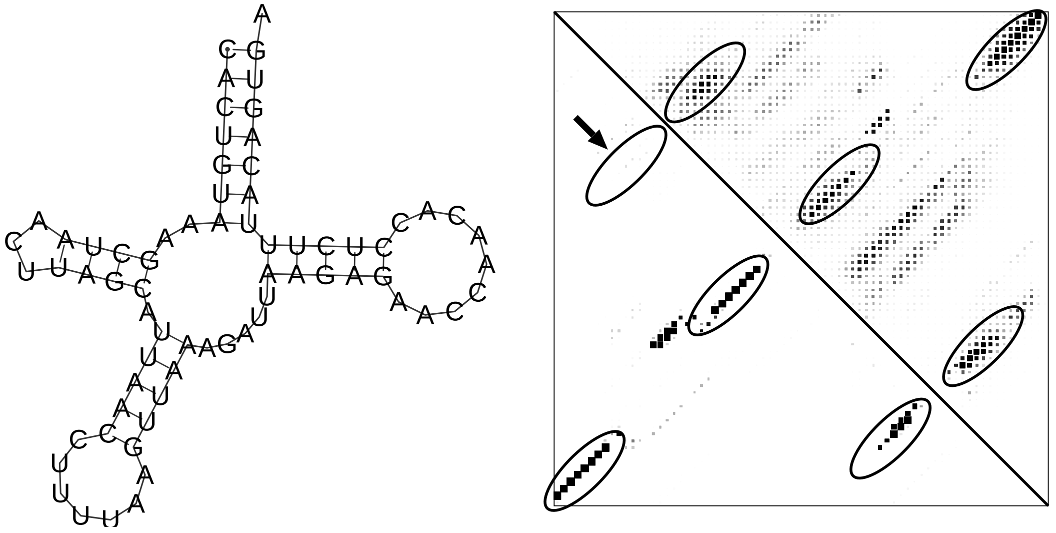


Figure 3.6. PCT for the base pairing probabilities. The left figure is the secondary structure of tRNA which is plotted using the RNAplot program of the Vienna RNA package [16]. The right figure illustrates the base pairing probabilities of a tRNA sequence. The lower left part of the matrix is computed by the McCaskill algorithm. The upper right part is after PCT. In both triangles, the region of the true stems of tRNA are indicated by ovals. The stem region that has been missed by the McCaskill algorithm is indicated by the arrow.

Table 3.4. Effects of PCTs on the accuracy of the alignments. The first column of each row indicates to which of the probabilities $p^{(a)}$ and $p^{(b)}$ the transformation is applied. The test set is the same as that of Table 3.1. For each accuracy measure, the highest value is shown in bold type face.

	SPS	MCC	SSS	SQS	PCS
$p^{(a)}$ and $p^{(b)}$	**0.81**	**0.71**	**0.81**	**0.79**	**0.55**
$p^{(b)}$	0.80	0.68	0.79	0.77	0.51
$p^{(a)}$	0.74	0.67	0.76	0.72	0.44
none	0.74	0.68	0.78	0.73	0.45

3.5. Summary

I have developed an efficient method to align multiple sequences of structural RNAs. The method first computes the base pairing probabilities and match probabilities. A simple Sankoff algorithm then is applied to obtain the final alignment by using these probabilities. Our scoring system enables us to incorporate complex secondary structure and homology information without complicating the Sankoff algorithm. I have shown that our method has the highest accuracy among the examined programs in terms of both the alignment quality and accuracy of structure prediction from the alignment. Our algorithm includes an efficient method to reduce the DP region to be computed, and this allows the alignment of long RNA sequences.

I have only optimized the proportionality constants of the loop match score and the stem match score and have not optimized the pair substitution matrix and the parameters of the models that are used to calculate the match and pair probabilities. It will be interesting to investigate the application of machine-learning methods in order to optimize these parameters in an integrated manner.

Chapter 4

Divide and Conquer Algorithm for Structural Alignment of RNA Sequences

4.1. Overview

In this chapter, I derive a divide and conquer algorithm for the Sankoff algorithm, which is analogous to the Hirschberg-Myers-Miller(H-M-M) algorithm for the sequence alignment. I first briefly describe the H-M-M algorithm, and then derive an analogous algorithm for a simple SCFG model. Finally, I describe the extension of the algorithm to the Sankoff model.

4.2. Hirschberg-Myers-Miller Algorithm

A divide and conquer algorithm for computing maximal common subsequences from a pair of sequences was first proposed by Hirschberg in the computer science literture [15], and was introduced into computational biology by Myers and Miller [32]. This algorithm does not use the memory for the traceback pointers and recursively determines the midposition of the alignment path using dynamic programming procedures similar to the forward and backward algorithms described in Algorithm 2. The space complexity of the algorithm is $\mathcal{O}(L)$ rather than $\mathcal{O}(L^2)$. The intuitive derivation of the Hirschberg-Myers-Miller (H-M-M) algorithm is as follows. For the mid-row i of the DP matrix, there must exist a state in the alignment path that emits the base $x_i^{(1)}$ at the row i. The state h and the column j at which the state h emits the base $x_i^{(1)}$ are computed by the following formula,

$$(j,h) = \mathrm{argmax}_{j',h',\Delta_{\mathrm{row}}(h')=1} F_{h'}(i,j') + B_{h'}(i,j') \tag{4.1}$$

where $F_h(i,j)$ is the maximal DP score of the alignment that aligns the subsequences $[1..i]$ and $[1..j]$ of sequences $x^{(1)}$ and $x^{(2)}$ respectively, and ends with the state h at the postion (i,j) of the DP matrix. $B_h(i,j)$ is defined such that $F_h(i,j) + B_h(i,j)$ is the maximal DP score of the alignment path that pass (i,j) with state h. $F_h(i,j)$ and $B_h(i,j)$ is computed by the algorithms, termed Viterbi-Forward and Viterbi-Backward, respectively, which are shown in Algorithm 4.2. These algorithms are similar to the forward and backward algorithms described in Algorithm 2, except that the recursions of the dynamic programming do not take the maximum of the left hand side in Algorithm 2, rather than take the sum in Algorithm 2. Once the state h and the column j is determined for the row i, then the DP regions that remains to be computed shrink to the diagonal blocks in the DP matrix asscociated with the alignment of subsequences $x_1^{(1)} \dots x_{i-1}^{(1)}$ and $x_1^{(2)} \dots x_{j-1}^{(2)}$, and that of subsequences $x_{i+1}^{(1)} \dots x_{L^{(1)}}^{(1)}$ and $x_{j+1}^{(2)} \dots x_{L^{(2)}}^{(2)}$ (Figure 4.2). Then, the same procedure is applied to the mid-rows of the diagonal blocks. The procedure is recursively applied until the column positions that emit the bases of sequence $x^{(1)}$ are determined. Finally, the alignment result is obtained by connecting the partial alignment paths with the states that emit only the bases of $x^{(2)}$ using the standard Viterbi algorithm. For the computation of $F_h(i,j)$, only the values of F_h on the neighboring cells $(i-1,j)$, $(i-1,j-1)$ and $(i,j-1)$ are needed. Similarly, to calculate the value of $B_h(i,j)$ only the values of neighboring cells are needed. Therefore, only the linear memory is reuired for the computation of Equation 4.1 (see Figure 4.1).

Algorithm 3 Viterbi-Forward and Viterbi-Backward algorithm.

Forward:

$$F_M(i,j) = s(i,j) + \max \begin{cases} F_M(i-1,j-1) \\ F_I(i-1,j-1) - d \\ F_D(i-1,j-1) - d \end{cases}$$

$$F_I(i,j) = \max \begin{cases} F_M(i,j-1) - d \\ F_I(i,j-1) - e \end{cases}$$

$$F_D(i,j) = \max \begin{cases} F_M(i-1,j) - d \\ F_D(i-1,j) - e \end{cases}$$

Backward:

$$B_M(i,j) = \max \begin{cases} B_M(i+1,j+1) - s(i+1,j+1) \\ B_I(i,j+1) - d \\ B_D(i+1,j) - e \end{cases}$$

$$B_I(i,j) = \max \begin{cases} B_M(i+1,j+1) + s(i+1,j+1) - d \\ B_I(i,j+1) - e \end{cases}$$

$$B_D(i,j) = \max \begin{cases} B_M(i+1,j+1) + s(i+1,j+1) - d \\ B_D(i+1,j) - e \end{cases}$$

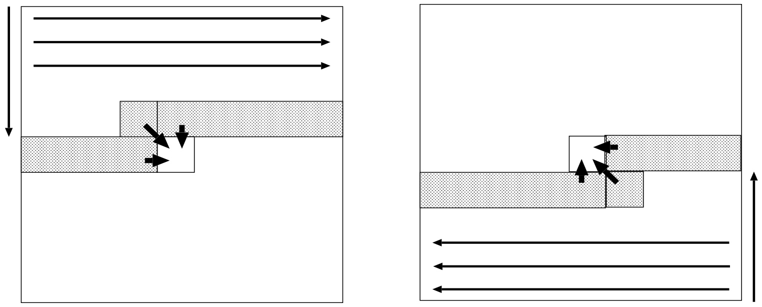
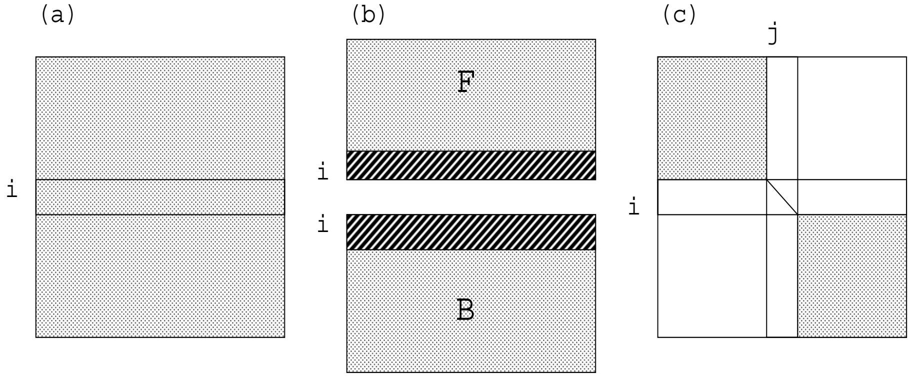


Figure 4.1. Procedures for Viterbi-Forward and Viterbi-Backward calculation.

Figure 4.2. Myers-Miller algorithm



4.3. Divide and Conquer Algorithm for Secondary Structure Prediction

An analog of the H-M-M algorithm in the case of SCFG models are derived similarly to the derivation in the previous section. In this section, I consider a simple SCFG model which has 5 states B, P, L, R and S as nonterminal symbols, and is defined by the following grammer.

$$\begin{aligned} B &\Longrightarrow SS \\ P &\Longrightarrow aPb|aLb|ab \\ L &\Longrightarrow aB|aL|a \\ R &\Longrightarrow Ba|Ra|a \\ S &\Longrightarrow B|P|L|R \end{aligned}$$

where a and b are terminal symbols which are base characters of the RNA sequence. Let k be the midpoint of the sequence x of length L. There exists a node in the tree that emits the base x_k.

If x_k is emitted as a left emission from the nonterminals, there exists a node (k, j, h) which corresponds to the parse subtree rooted at the nonterminal h for subsequence $x_k \ldots x_{j-1}$. The position j and the state h is obtained by the formula,

$$(i, h) = \text{argmax}_{i'<k,h',\Delta_r(h')=1} F_{h'}(i', k) + B_{h'}(i', k)$$

where the $F_h(i, j)$ is the maximal DP score of the subtrees which are rooted at nonterminal h for subsequence $x_i..x_{j-1}$. $B_h(i, j)$ is defined such that $F_h(i, j) + B_h(i, j)$ is the maximal score of the parse trees which pass through the point (i, j) in the DP matrix with state h. $F_h(i, j)$ and $B_h(i, j)$ is calculated by the algorithms similar to the inside and outside algorithms, These algorithms are called the CYK-Inside and CYK-Outside algorithms in this theses and defined by, where $s((i, j), (i + 1, j - 1))$ is the score assigned to the stacking base pairs (x_i, x_j) and (x_{i+1}, x_{j-1}). $\Delta_l(h)$ and $\Delta_r(h)$ is the number of left and right emission of the state h, which is defined in Table 4.1.

If x_k is emitted as a right emission, there exists a node (i, k, h) in the parse tree which is associated to the parse subtree rooted at nonterminal h for subsequence

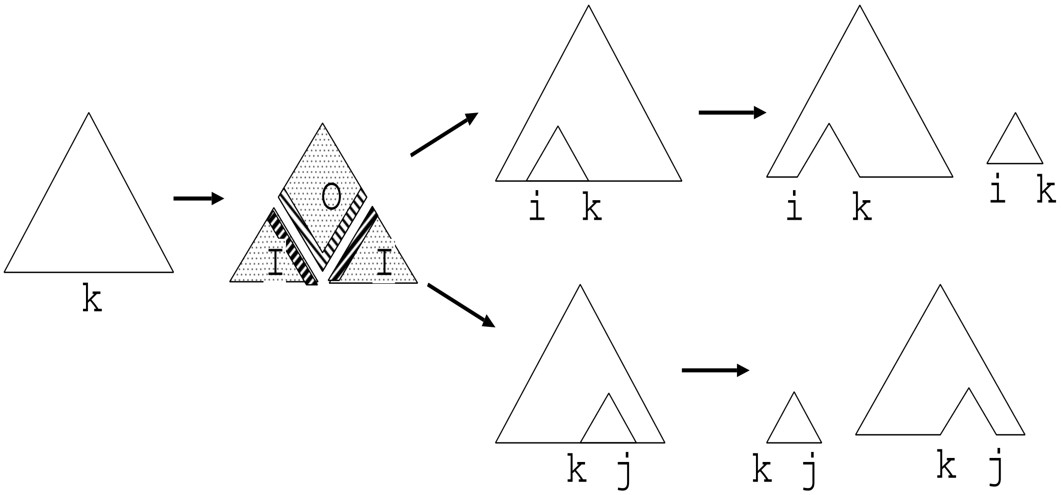


Figure 4.3. Divide and Conquer algorithm for trinangle-shaped DP region.

Table 4.1. The number of bases emitted by the states of SCFG

state	Δ_l	Δ_r
P	1	1
L	1	0
R	0	1
B	0	0
S	0	0

Algorithm 4 CYK-Inside and CYK-Outside algorithm.

CYK-Inside:

$$F_P(i,j) = \max \begin{cases} F_P(i+1,j-1) + s((i,j),(i+1,j-1)) \\ F_R(i+1,j-1) \\ F_L(i+1,j-1) \\ F_B(i+1,j-1) \end{cases}$$

$$F_R(i,j) = \max_{h=P,R,L,B} F_h(i,j-1)$$

$$F_L(i,j) = \max_{h=P,L,B} F_h(i+1,j)$$

$$F_S(i,j) = \max_{h=P,R,L,B} F_h(i,j)$$

$$F_B(i,j) = \max_{i \leq k \leq j} [F_h(i,k) + F_h(k,j)]$$

CYK-Outside:

$$B_P(i,j) = \max \begin{cases} B_P(i-1,j+1) + s((i-1,j+1),(i,j)) \\ \max_{h=R,L,S} B_h(i-\Delta_l(h), j+\Delta_r(h)) \end{cases}$$

$$B_R(i,j) = \max_{h=P,R,S} B_h(i-\Delta_l(h), j+\Delta_r(h))$$

$$B_L(i,j) = \max_{h=P,R,L,S} B_h(i-\Delta_l(h), j+\Delta_r(h))$$

$$B_B(i,j) = \max_{h=P,R,L,S} B_h(i-\Delta_l(h), j+\Delta_r(h))$$

$$B_S(i,j) = \max \begin{cases} \max_{j \leq k \leq L} [B_S(i,k) + F_S(j,k)] \\ \max_{1 \leq k \leq i} [F_S(k,i) + B_S(k,j)] \end{cases}$$

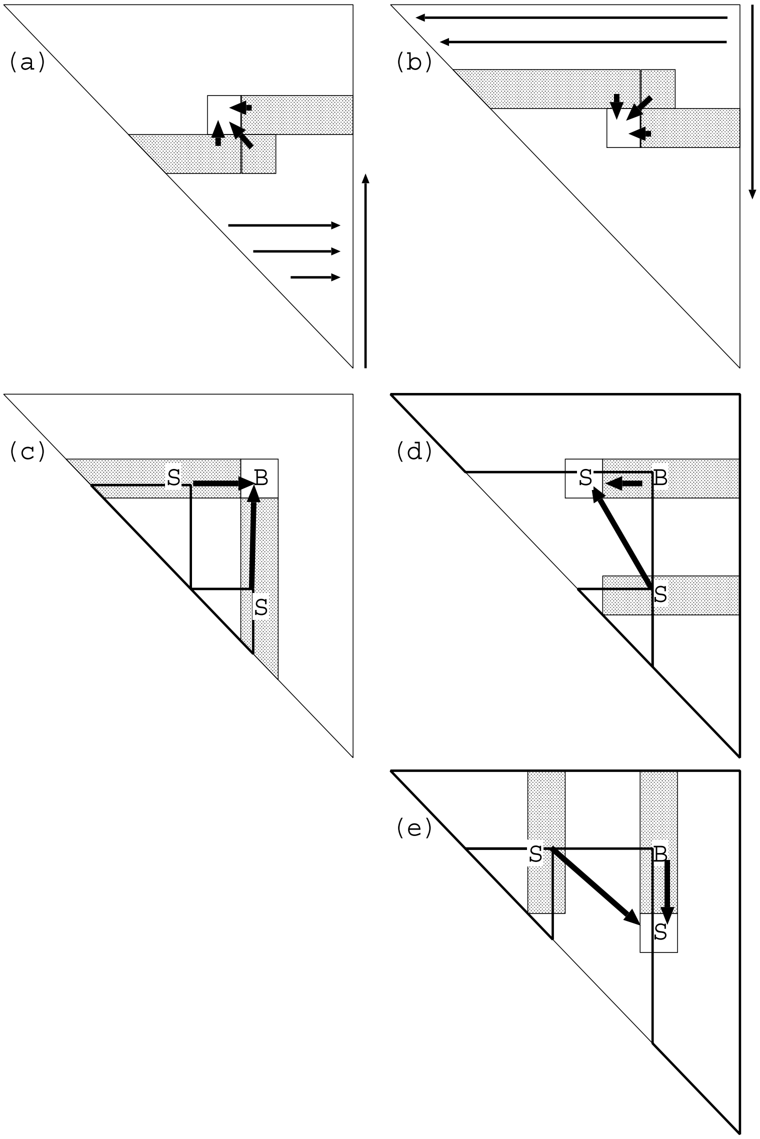


Figure 4.4. Procedures for CYK-Inside and CYK-Outside computaion.

$x_i \dots x_{k-1}$. The position i and the state h is computed by following formula,

$$(j, h) = \text{argmax}_{k<j',h',\Delta_l(h')=1} F_{h'}(k, j') + B_{h'}(k, j')$$

The true node which emits the base x_k is given by the node which corresponds to the larger DP score.

Once the node (i, j, h) on the parse tree is determined, the trianglular DP region that remains to be computed splits into two pieces: a trianglular region associated to the subsequences $\{x_{i'} \dots x_{j'} | i \leq i' \leq j' < j\}$ and a caret-shaped region associated to the subsequences $\{x_{i'} \dots x_{j'} | 1 \leq i' < i, j \leq j' < L\}$.

For the triangular region, The same procedure is applied to the midpoint of the segment $x_i \dots x_{j-1}$. For the caret-shaped region, the midpoint k of the longer segment of $x_1 \dots x_{i-1}$ and $x_j \dots x_L$ is considered. As in the case of the triangular region, the node in the parse tree the emits x_k is obtained by the CYK-Inside and CYK-Outside computations. Depending on the position of the node in the DP region, there are three possible cases where the caret-shaped DP region splits into smaller regions. In the first case, the region splits into two caret-shaped regions. In this case, the same procedures described above is applicable. In the other two cases, the DP region splits into a triangular region which is treated in the same way as described previously, and a nose-shaped region. For the nose-shaped region, there exists a unique node (i, j, B) in the parse tree which bifurcate into two subtrees rooted at (i, k, S) and (k, j, S). for some k. Here, the sequence positions i, k and j belong to the left, mid and right segments of the nose-shaped DP region, respectively, which are obtained by the following formula.

$$(i, k, j) = \text{argmax}_{i',k',j'} B_B(i', j') + F_S(i', k') + F_S(k', j')$$

After the three nodes are determined, the nose-shaped DP region splits into three pieces of caret-shaped regions. Since these regions are treated by the previous procedures, the whole parse tree can be recursively determined. Although the algorithm does not need the memory for the traceback pointers, as in the case of the H-M-M algorithm, the computation of $F_B(i, j)$ and $B_S(i, j)$ need the two-dimensional memories to store the values of $F_S(i, j)$ and $B_B(i, j)$, hence the space complexity of the algorithm is the same $\mathcal{O}(L^2)$ as the standard CYK algorithm. The proposed algorithm has some similarity with the divide and conquer

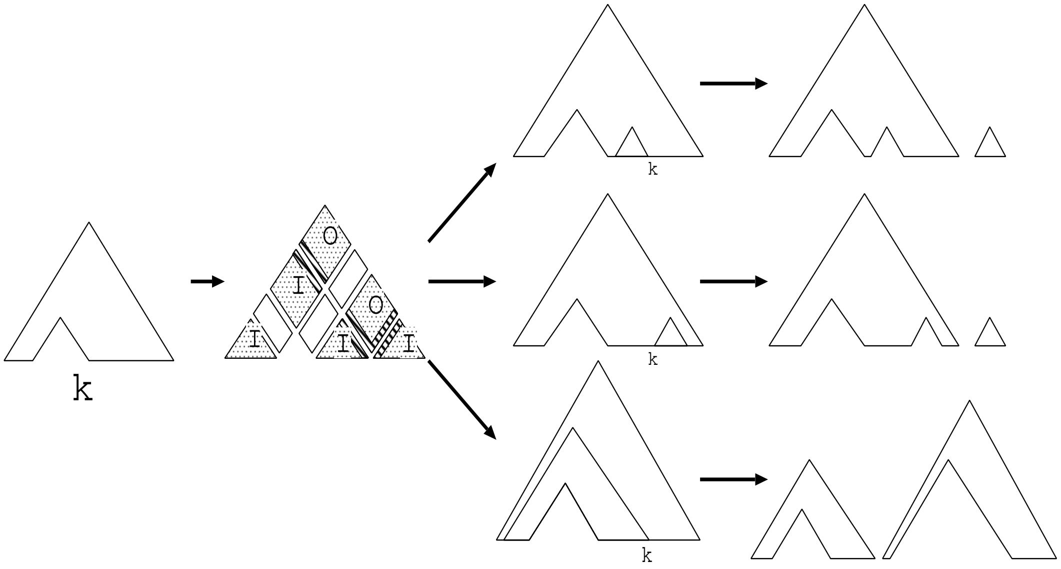


Figure 4.5. Divide and Conquer algorithm for caret-shaped DP region.

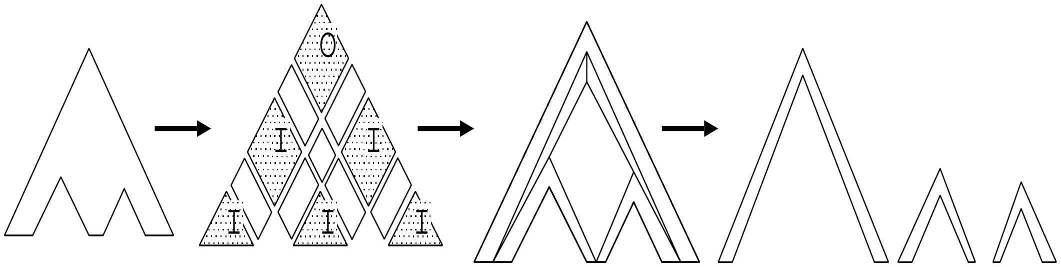


Figure 4.6. Divide and Conquer algorithm for nose-shaped DP region.

algorithm for the profile-SCFG model proposed by S. Eddy in [9]. The present algorithm is different from his approach since the divisions of the computations into smaller pieces are performed in the sequence dimensions rather than in the dimension of the states of the model.

4.4. Divide and Conquer Algorithm for Structural Alignment

It is straightforward to extend the divide and conquer algorithm derived in the previous section to the Sankoff model. Here we consider a variant of the Sankoff model that have nine nonterminals and defined by the following grammer,

$$\begin{aligned}
B &\Longrightarrow SS \\
P &\Longrightarrow a^{(1)}a^{(2)}Sb^{(1)}b^{(2)}|a^{(1)}a^{(2)}b^{(1)}b^{(2)} \\
ML &\Longrightarrow a^{(1)}a^{(2)}S|a^{(1)}a^{(2)} \\
IL &\Longrightarrow a^{(2)}S|a^{(2)} \\
DL &\Longrightarrow a^{(1)}S|a^{(1)} \\
MR &\Longrightarrow Sb^{(1)}b^{(2)}|b^{(1)}b^{(2)} \\
IR &\Longrightarrow Sb^{(2)}|b^{(2)} \\
DR &\Longrightarrow Sb^{(1)}|b^{(1)} \\
S &\Longrightarrow B|P|ML|IL|DL|MR|IR|DR
\end{aligned}$$

where $a^{(h)}$ and $b^{(h)}$ are the terminal symbols associated to the bases of the sequence $x^{(h)}$.

The triangular, caret-shaped, and nose-shaped DP regions in the previous section have direct analogs in the Sankoff model. For example, a triangular DP region is defined for a pair of subsequences $x^{(1)}_{i_0} \ldots x^{(1)}_{i_1}$ and $x^{(2)}_{j_0} \ldots x^{(2)}_{j_1}$, by the set

$$\{(i, j, k, l)|i_0 \le i \le k \le i_1, j_0 \le j \le l \le j_1\}.$$

The caret-shaped region and the nose-shaped region are similarly defined such that each sequence segment parsed by the DP region in the SCFG case are replaced with a pair of sequence segments.

For the triangular and the caret-shaped regions, the mid-row r os the current DP region is considered. If the base $x_r^{(1)}$ is emitted as a left emission, the corresponding node of the parse tree is obtained by the equation

$$(j,k,l,h) = \text{argmax}_{j,k,l,h,\Delta_l^{(1)}(h)=1} F_h(r,j,k,l) + B_h(r,j,k,l)$$

where $\Delta_l^{(1)}(h)$ is the number of the emitted base left of the nonterminal h with respect to the sequence $x^{(1)}$. $F_h(i,j,k,l)$ and $B_h(i,j,k,l)$ are calculated by the CYK-Inside and CYK-Outside algorithm of the Sankoff model. In the case where $x_r^{(1)}$ is emitted as a right emission, the corresponding emission node is similary calculated. Once the emission node is obtained, the region is split into smaller pieces.

For the nose-shaped DP region, there exists are unique bifurcation node (i,j,k,l,B) and two child nodes (i,j,u,v,S), $(u,v,k.l,S)$ These nodes are calculated by the formula,

$$(i,j,u,v,k,l) = \text{argmax}_{i,j,u,v,k,l} \left[B_B(i,j,k,l) + F_S(i,j,u,v) + F_S(u,v,k,l) \right]$$

After the determination of the three nodes, the DP region splits into three pieces of caret-shaped regions.

These procedures are applied recursively until all the nodes that emit the bases of sequence $x^{(1)}$ are determined. Finally, the whole parse tree is constructed by complementing the nodes of no emission and the nodes which emit only the bases of the second sequence $x^{(2)}$, using the standard CYK algorithm for the Sankoff model.

As in the case of SCFG model, the algorithm does not require the memory for the traceback pointers. However, the space complexity of computation is $\mathcal{O}(L^4)$, since the calculation of $F_B(i,j,k,l)$ and $B_S(i,j,k,l)$ require the four dimensional memories to store the values of $F_S(i,j,k,l)$ and $B_B(i,j,k,l)$. The computation of the other states only require the three dimensional memories to store neighboring values of $F_h(i,j,k,l)$ and $B_h(i,j,k,l)$.

4.5. Use of New Algorithm

In subsection 3.3.3, I introduced the skip approximation which constrains the bifurcation computation to a restricted set of positions in the DP matrix. When the skip approximation is combined with the divide and conquer algorithm, the $\mathcal{O}(L^4)$ part of the memory reduces by $1/\kappa^4$ times for skip size κ, since the only the memories for the bifurcation calculation need to be forth order in length L. Therefore, the algorithm is useful for aligning long RNA sequences that cannot be aligned by the standard CYK algorithm due to the limitation of available memory.

Chapter 5

Robust prediction of consensus secondary structures using averaged base pairing probability matrices

5.1. Overview

In this chapter, I propose an algorithm that predicts the consensus secondary structures from the alignments which is robust against alignment failures. Firstly, I describe the programs that are frequently used to predict the consensus secondary structures from multiple alignments. I also describe three algorithms all of which maximize the expected accuracy of secondary structure candidates under different base pairing probability distributions. Then I show the result of the experiments that compare the accuracy of the consesus secondary structure prediction from the alignments that are created by human curation and by computer programs. I show that one of the algorithms, termed McCaskill-MEA, is the robustest against alignment failures than the others. The McCaskill-MEA method performs especially better than others, for the low quality alignments and the alignments that consists of many sequences. The model has a parameter that controls the sensitivity and specificity of predictions. I discuss the uses of the parameter for multi-step screening procedures to search for conserved secondary structures, and for assigning confidence values to the predicted base pairs.

5.2. Background

Since the existence of conserved secondary structures among phylogenetic relatives indicates the functional importance of such transcripts, several research groups have sought for conserved secondary structures on a genomic scale [20],[44],[43],[35].

In their studies, a large number of multiple alignments were created using computer programs, and consensus secondary structures were then predicted from these alignments. They used alignment programs that neglected the special conservation patterns of secondary structures such as the base covariations in the stem regions since the alignment algorithms that took into account the base covariations required huge computational resources [38], [17],[28]. Therefore, there were potential risks of overlooking conserved secondary structures due to misalignments. Such loss of sensitivity is particularly problematic in the early stage of large-scale screening that precedes the time-consuming but accurate computational and experimental validation stages.

In this chapter, I investigate the dependence of the accuracy of secondary structure prediction on the quality of alignments and propose a method to predict conserved secondary structures from multiple alignments, which is robust against alignment failures. Our algorithm first computes the base pairing probability matrix for each sequence in the alignment and then obtains the base pairing probability matrix of the alignment by averaging over these matrices. The consensus secondary structure is predicted from this matrix using a Nussinov-style dynamic programming algorithm [33].

The use of the average pair probability matrix for obtaining the consensus structures is not a new idea and there have been several studies [21],[18],[25],[23],[26] that have used the base pairing probability matrices of single sequences to predict the consensus secondary structures. In particular, the ConStruct program [26],[25] predicts the same consensus structures as those predicted by our algorithm with a specific parameter value. However, I present a new interpretation and justification of this method in terms of the maximal expected accuracy

(MEA) principle [31], which has been successfully applied recently to the sequence alignment and the structure prediction to single sequences. This new interpretation makes it obvious that the method has an advantage in predicting the structures from seriously misaligned sequences. I show that our algorithm outperforms the leading programs for the consensus structure prediction [19],[24] in such a situation.

5.3. Methods

5.3.1 Algorithms for Consensus Structure Prediction

For a given alignment of length L, which is composed of N sequences X, let $\mathcal{C} = \{I|1 \leq I \leq L\}$ be the set of positions of alignment columns and let $\mathcal{PC} = \{(I,J) \in \mathcal{C} \times \mathcal{C}|1 \leq I < J \leq L\}$. The consensus secondary structure $y^{(b)}$ of the alignment is defined as in the case of single sequences (2.1.1).

Two programs RNAAlifold and Pfold are often used for predicting the consensus secondary structures from given multiple alignments. RNAAlipfold [19] is a multi-sequence extension of the McCaskill algorithm. For each consensus secondary structure candidate, it assigns a Boltzmann factor,

$$P(y|X) = \frac{1}{Z(X)} \exp\left(-\frac{E(y,X)}{kT} + \mathrm{Cov}(y,X)\right)$$

where $E(y,X)$ is the mean energy of the secondary structures of sequences X, all of which are assumed to form the same structure y, and $\mathrm{Cov}(y,X)$ is the base covariation bonus factor that gives a positive value for stem-conserving covariations. I consider an MEA algorithm that uses the base pairing probability matrices as calculated by RNAAlipfold and refer to it as RNAAlipfold-MEA. The maximal likelihood version of the RNAAlipfold algorithm corresponds to RNAAlifold [19] and is compared with other programs in the following section.

Pfold [24] is a multi-sequence extension of the SCFG model for a single sequence structure prediction. The differences from the single sequence case are that it simulteneously emits bases in each column and each pair of columns, and that the emission scores assume the likelihood values computed by the Markov model of sequence evolution. The Pfold algorithm is an MEA algorithm that maximizes the expected accuracy with $\alpha = 1$.

Both RNAAlipfold and Pfold assume the correctness of alignments, and the covariation scores contained in both the models rely on it. Their covariation scores are most efficient when they are applied to high-quality multiple alignments. However, in low-quality alignment data, there are many *fake* inconsistent mutations caused by alignment failures, which may cause the incorrect estimations of the covariation scores and result in the loss of sensitivity to conserved structures.

Here, I propose an alternative MEA algorithm that is not strongly dependent on the correctness of alignments. First, I define the conditional probability distribution function $P(Y|X)$ over all the secondary structures of all the sequences in the alignment as follows:

$$P(Y|X) = \prod_{x \in X} P(y^{(x)}|x)$$

where $Y = \{y^{(x)}|x \in X\}$ denote the set of secondary structures of sequences X, and $P(y^{(x)}|x)$ is given by the Boltzmann distribution of single sequence x (Equation (2.7)). For each consensus structure candidate $\mathcal{S} = \{\mathcal{L}, \mathcal{P}\}$, I define the expected accuracy $EA_\alpha(\mathcal{S})$ of the structure as the mean value of the expected accuracies of sequences.

$$\begin{aligned} EA_\alpha(\mathcal{S}) &= E\left[\frac{1}{N}\sum_{x \in X}\left\{\alpha \sum_{i \in \mathcal{L}} \prod_{j \neq i} \delta(y_{ij}^{(x)}, 0) + 2 \sum_{(i,j) \in \mathcal{P}} \delta(y_{ij}^{(x)}, 1)\right\}\right] \\ &= \alpha \sum_{i \in \mathcal{L}} q_i + 2 \sum_{(i,j) \in \mathcal{P}} p_{ij} \end{aligned}$$

where p_{ij} is the mean value of the base pairing probabilities $p_{ij}^{(x)}$,

$$
\begin{aligned}
p_{ij} &= E\left[\frac{1}{N}\sum_{x' \in X} \delta(y_{ij}^{(x')}, 1)\right] \\
&= \sum_{y^{(x_1)}}\sum_{y^{(x_2)}}\cdots\sum_{y^{(x_N)}} \left(\frac{1}{N}\sum_{x' \in X} \delta(y_{ij}^{(x')}, 1)\right) P(Y|X) \\
&= \frac{1}{N}\sum_{x \in X}\sum_{y^{(x)}} \delta(y_{ij}^{(x)}, 1) P(y^{(x)}|x) \\
&= \frac{1}{N}\sum_{x \in X} p_{ij}^{(x)} \qquad (5.1)
\end{aligned}
$$

and q_i is given by Equation (2.17).

The MEA structure is computed from q_i and p_{ij} in a manner identical to the case of the prediction from single sequences. I refer to the algorithm as McCaskill-MEA. For $\alpha = 0$, the McCaskill-MEA algorithm predicts the same structures as those of the ConStruct program [26],[25].

The McCaskill-MEA algorithm does not assume that all the sequences take an equal single structure and instead predicts the structure that is supported by the majority of sequences. Since the model does not include any covariation score term, the accuracy of the prediction may be lower than that of other algorithms for high-quality alignments. However, McCaskill-MEA has the advantage that the algorithm is free from the negative effects of covariation scores in the presence of severe alignment errors.

To observe the effect of the suboptimal structures contained in the base pairing probability matrices $p_{ij}^{(x)}$, I consider another MEA algorithm. It first computes the predicted structures for all sequences using the Mfold program and defines the base pairing probability matrix of the sequence as

$$
p_{ij}^{(x)} = \begin{cases} 1 & (i,j) \text{ is predicted to form a base pair by Mfold} \\ 0 & \text{otherwise} \end{cases}
$$

The base pairing probabilities of the alignment are computed as shown in Equation (5.1), and the predicted structure is computed by Equation (2.9). The algorithm is referred to as Mfold-MEA.

I used version 1.5 of the Vienna RNA package [16] for the computation of the base pairing probability matrices of McCaskill-MEA and RNAalipfold-MEA and the structure predictions of the RNAAlifold algorithm. Version 5 of Mfold was used for the computation of the Mfold-MEA algorithm. A stand-alone program of Pfold was obtained (courtesy of Dr. B. Knudsen).

5.4. Results

5.4.1 Comparison of Algorithms

Figure 5.1 shows examples of the density plots of the base pairing probabilities, which are calculated from a multiple alignment of ten tRNA sequences. The alignment is created using the ClustalW software. The lower left triangles in both the figures show the *true* distribution of base pair probabilities. They are computed by first assigning the annotated structure in the Rfam database to each sequence and then computing base pairing probability matrices in a manner similar to Mfold-MEA. Although the true tRNA structure has only four stems (Figure 5.1 bottom), about ten stems are observed in the plot due to severe misalignments. The upper right triangles show the density plot of the pairing probabilities used in the RNAAlipfold-MEA (left) and McCaskill-MEA (right) algorithms. Only two out of four stems are observed in the matrix for RNAAlipfold-MEA, while all the four stems are observed in the matrix for McCaskill-MEA.

Figure 5.2 shows the receiver operator characteristic (ROC) curves of the structure predictions from alignments of ten sequences. The x and y axis represent the specificity and the sensitivity of predictions, respectively. The ROC curves are computed by varying α in the three MEA algorithms. The sensitivity is large for small values of α, since the terms that score the base pairs in the expected accuracy is emphasized and the number of predicted base pairs increases. The ROC curve reaches a limit for $\alpha \to 0$. In this limit, the entire regions of the multiple alignments are filled with predicted stems. For large values of α, the number of predicted base pairs decreases. In the limit of large α, the number of predicted stems is so small that the corresponding plot fluctuates due to statistical fluctuations. Therefore, I only showed the data points for which the total number

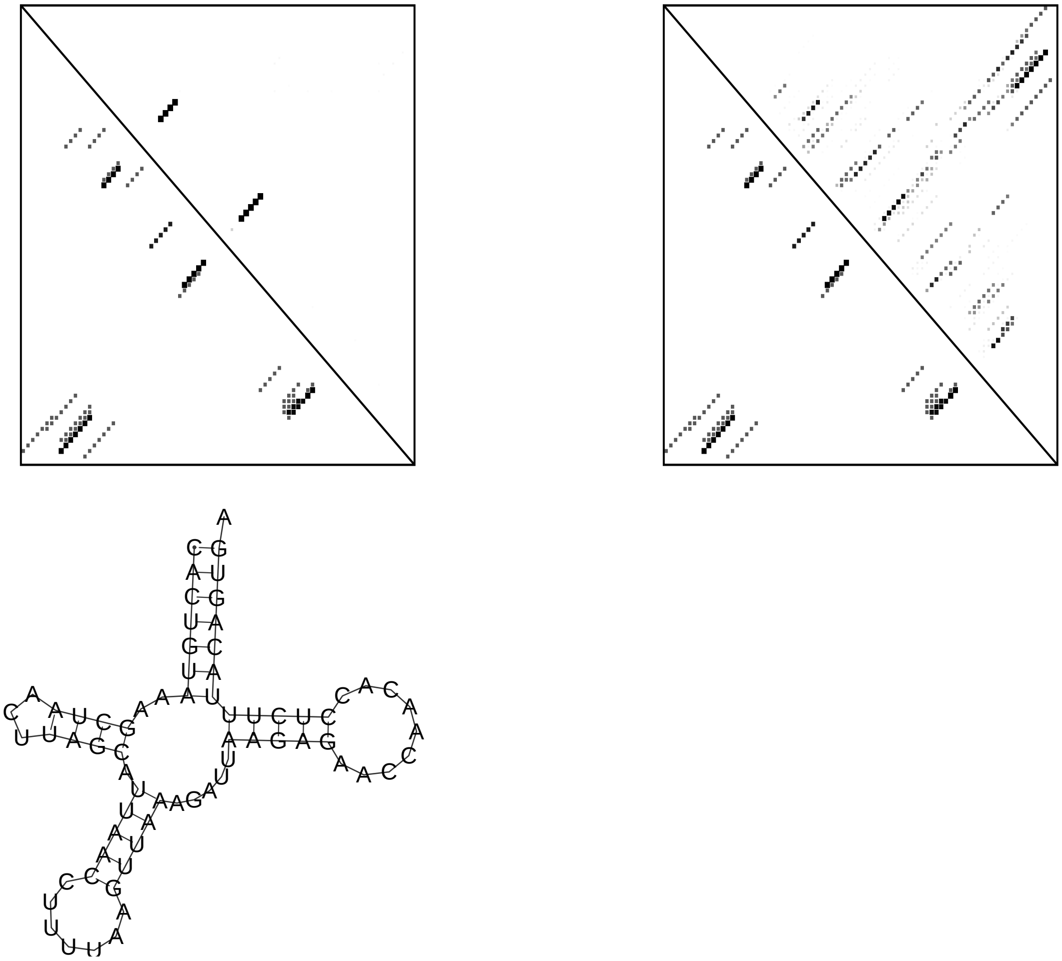

A
C G
A U
C G
U A
G C
U A
C A A U C G A A A U U U C U C C A C
U U A G C A A G A G A A C A
A U A A G A U
U A
A U
A U
C C G
U A
U A
U U

Figure 5.1. Density plots of base pairing probability matrices. In both the matrices, the lower left triangle is the *true* distribution of base pair probabilities that is derived from the Rfam annotation of the tRNA secondary structure. The upper right triangles are the base pairing probabilities used in the RNAAlipfold-MEA and McCaskill-MEA algorithms respectively. The true tRNA secondary structure is shown in the bottom figure, which is plotted using the RNAplot program of Vienna RNA package [16].

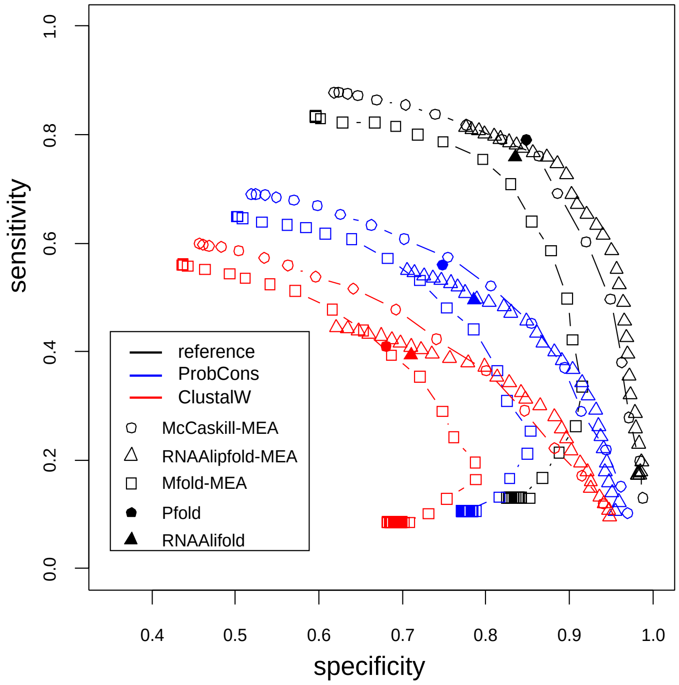


Figure 5.2. ROC plot of the consensus structure predictions. The x and y axes represent the specificity and sensitivity of predictions, respectively. The colors indicate the types of alignments from which the consensus structures are predicted. The black, blue, and red colors correspond to the reference alignments of the Rfam database, the ProbCons alignments, and the ClustalW alignments, respectively. The character symbols indicate the types of structure prediction algorithms: McCaskill-MEA (open circle), RNAAlipfold-MEA (open triangle), Mfold-MEA (open square), Pfold (filled circle), and RNAAlifold (filled triangle). For McCaskill-MEA, RNAAlipfold-MEA and Mfold-MEA, multiple points are computed by varying the parameter α, and their trajectories are connected by lines.

Table 5.1. The ROC score, maximal MCC, and maximal sensitivity for each alignment type and structure prediction algorithm. Since I can obtain only one point in the sensitivity-specificity plane for Pfold and RNAAlifold, I cannot show the ROC score for these softwares. Further, the maximal MCC and maximal sensitivity is the MCC and sensitivity at that point for these softwares. For other algorithms, the ROC score is defined as the area of the convex region that is spanned by the data points and the points $\{(0,0),(\mathrm{sp}_{\mathrm{max}},0),(0,\mathrm{sn}_{\mathrm{max}})\}$ in the specificity-sensitivity plane, where $\mathrm{sp}_{\mathrm{max}}$ and $\mathrm{sn}_{\mathrm{max}}$ denote the maximal specificity and sensitivity of the data points, respectively.

Alignment	Algorithm	ROC score	Max MCC	Max sensitivity
Reference	McCaskill-MEA	**0.81**	0.81	**0.88**
	RNAAlipfold-MEA	0.77	0.81	0.81
	Mfold-MEA	0.73	0.77	0.83
	Pfold	-	**0.82**	0.79
	RNAAlifold	-	0.80	0.76
ProbCons	McCaskill-MEA	**0.60**	**0.66**	**0.69**
	RNAAlipfold-MEA	0.50	0.63	0.55
	Mfold-MEA	0.51	0.62	0.65
	Pfold	-	0.65	0.56
	RNAAlifold	-	0.62	0.50
ClustalW	McCaskill-MEA	**0.48**	**0.57**	**0.60**
	RNAAlipfold-MEA	0.39	0.54	0.44
	Mfold-MEA	0.40	0.54	0.56
	Pfold	-	0.53	0.41
	RNAAlifold	-	0.53	0.39

of predicted base pairs is greater than ten percent of the total number of true base pairs. Since there is no parameter to control the specificity-sensitivity trade-off for Pfold (filled circle) and RNAAlifold (filled triangle), only one point for each alignment type is plotted.

Table 5.2 shows the p values of the Fisher's sign test that tests the null hypothesis that McCaskill-MEA is not more accurate than each of the other programs. The first column shows the name of the program compared. The other columns show the p values that measure the unlikeliness of the null hypothesis. The number of observations that the accuracy of Murlet is better (+) or worse (-) than that of the program being compared is also shown in the round brackets. The table indicates that the McCaskill-MEA method is comparable to the leading programs of consensus structure prediction even for the reference alignments. For the alignment set created by the alignment programs ProbCons and ClustalW, McCaskill-MEA performs better than these programs with p-values ranges from 7e-3 % to 9.4 %.

Figure 5.2 shows that the sensitivity considerably depends on the alignment quality; the maximal sensitivity achieved for ClustalW alignments is less than ProbCons alignments by 10 percent and less than the reference alignments by about 30 percent. For all alignment types, the curves of McCaskill-MEA are above Mfold-MEA, which shows the efficiency achieved by including the effect of suboptimal structures in the consensus structure prediction. The higher sensitivity of RNAAlipfold-MEA as compared to that of RNAAlifold at the same specificity values also indicates the superiority of the MEA algorithm as compared to its maximal likelihood version, although the difference is less prominent. Both the specificity and sensitivity decrease for Mfold-MEA in the large α limit, which indicates that the loop probability values q_i incorrectly assume large values at the columns of the true base pairs due to the neglect of suboptimal structures. As expected, Pfold and RNAAlipfold-MEA show slightly better sensitivities for specificities larger than 0.8 as compared to McCaskill-MEA due to the positive effect of the base covariation scores. However, McCaskill-MEA shows the best sensitivities for lower specificity regions in all the three alignment types.

Table 5.1 lists the ROC score, the maximal MCC, and the maximal sensitivity for each alignment type and structure prediction algorithm. The ROC score is defined by the area under the ROC curve and is a standard indicator for prediction efficiency. Table 5.1 shows that the ROC score is the highest for the McCaskill-MEA algorithm. As for the maximal MCC, the Pfold program achieves the best MCC value for the high quality reference alignments. However, McCaskill-MEA is better than the RNAAlifold program even for these alignments. For the ProbCons and ClustalW alignments, McCaskill-MEA algorithm outperforms the other programs. The difference between the maximal MCC value of McCaskill-MEA and that of other algorithms is larger for the lower quality ClustalW alignments. The table also shows that the maximal sensitivity is the highest for the McCaskill-MEA algorithm.

Note that in contrast to the specificity, the sensitivity values cannot be arbitrarily close to 1 unless the model's accuracy is fairly high; this is because it is not possible to predict *all* the base pair candidates (i.e., $y_{ij} = 1$ for all $1 \leq i < j \leq L$) to satisfy the consistency constraint (Equation (2.2)). At this point, the secondary structure prediction problem is different from other binary classification problems where the classifier that predicts all the test samples as positive (which corresponds to predicting $y_{ij} = 1$ for all $1 \leq i < j \leq L$), trivially achieves a sensitivity of one. Therefore, it may be said that the maximal reachable sensitivity is itself an indicator of the efficiency of the algorithms, and that McCaskill-MEA is comparatively much better than other algorithms with respect to it.

Figure 5.3 shows the dependence of the ROC score on the number of sequences in the alignments. The alignments of 2, 4, 6, and 8 sequences are created by sampling sequences randomly from the alignments of ten sequences. The colors and symbols are the same as in Figure5.2. The ROC scores of McCaskill-MEA and Mfold-MEA increase with the number of sequences, while the increase of RNAAlipfold-MEA is somewhat slower than that of the other algorithms. For computationally aligned sequences, the McCaskill-MEA algorithm shows the best performance among the three algorithms. Even for the reference alignments, the McCaskill-MEA algorithm has a slightly better ROC score than the RNAAlipfold-MEA algorithm, which might imply the difficulty to score sequence covariations correctly for diverged sequences.

Figure 5.4 shows an example of the dependence of the sensitivity at fixed specificity (0.7) on the mean sequence identities. The current Rfam dataset has not permitted us to collect multiple sequence alignments over a wide range of sequence identities for various families. The used dataset consists of 188 multiple alignments of four sequences collected from the Rfam Hammerhead_3 ribozyme family. The alignments are categorized into bins of width ten percent in their mean sequence identities, and the mean sensitivities for each bin are plotted. The colors and symbols have the same meanings as in Figure 5.2 and 5.3. For

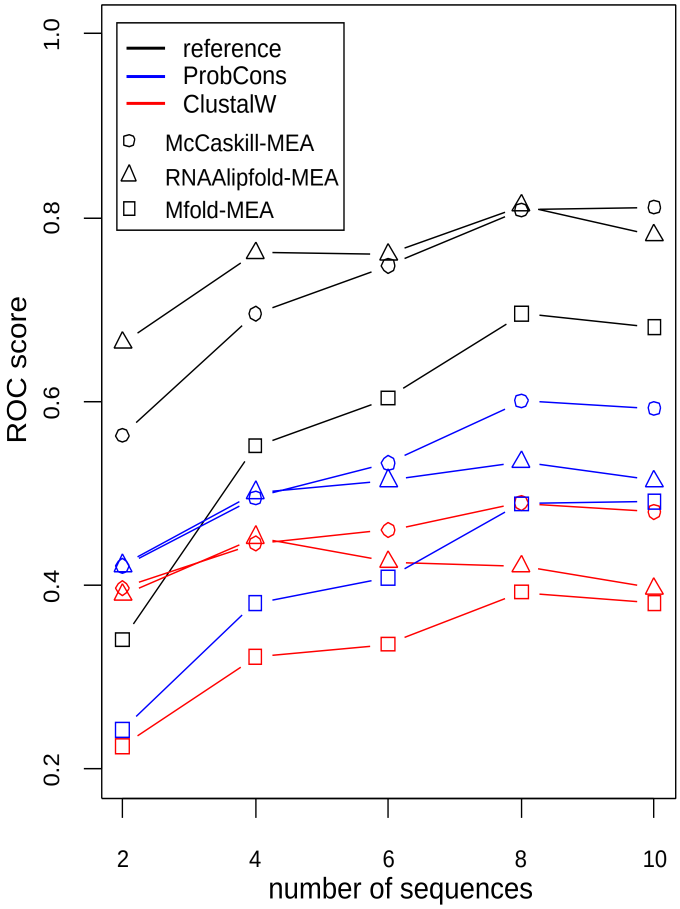


Figure 5.3. Dependence of the ROC score on the number of sequences in the alignments. The colors and symbols have the same meanings as in Figure 5.2.

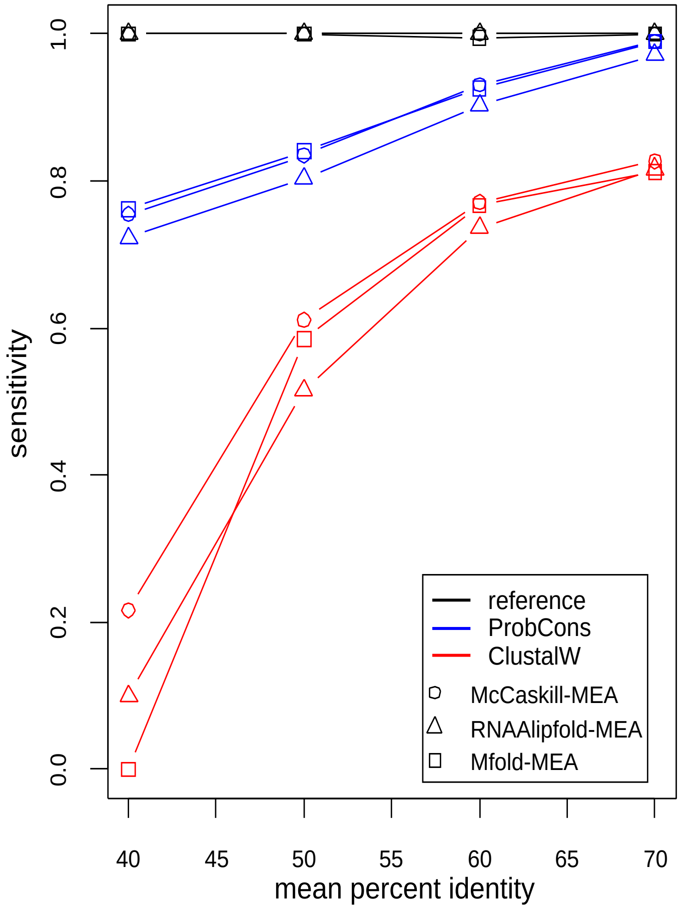


Figure 5.4. An example of the sequence identity dependence of sensitivity at a specificity of 0.7. The colors and symbols have the same meanings as in Figures 5.2 and 5.3. The dataset is taken from the Hammerhead_3 family of the Rfam database. The dataset consists of 188 multiple alignments of four sequences. They are binned according to their mean pairwise sequence identities, and the result is averaged over each bin.

this family, all three algorithms show similar behaviors for the reference alignments and the ProbCons alignments. For the ClustalW alignments, however, the McCaskill-MEA algorithm shows the best sensitivity in the region where the sequence identity is less than 60 percent.

5.4.2 Uses of Parameter α

As I have shown, the prediction accuracy of conserved secondary structures significantly depends on the alignment quality, which indicates the necessity of refining the alignments after candidates of alignments with conserved structures are screened. The parameter α, which controls the sensitivity and specificity of the prediction accuracy, can be conveniently used for such multi-step screening procedures; this is done by taking small values of α to screen conserved structure candidates from coarse alignments with high sensitivity and then taking α large to predict structures from refined multiple alignments with high specificity.

Another use of the parameter α is to assign confidence values to predicted base pairs. Figure 5.5 shows an example of the predicted structures of the UnaL2 family for varying parameter α. They are predicted from a ProbCons alignment using the McCaskill-MEA algorithm. As seen from the figure, the number of base pairs monotonically decreases with α without creating alternative base pairs. This behavior holds in most cases. Hence, I define the confidence value of each predicted base pair as follows. For any base pair, we associate the α value that is maximal among the ones whose MEA solutions predict that pair, and define the confidence value of the pair as the specificity corresponding to that α (Figure 5.6 (left)). The confidence value of a predicted base pair represents the empirical probability that the pair is a true base pair. The definition of confidence value depends only on the test dataset and the corresponding ROC curve and is essentially independent of the details of models. It has the property that the number of base pairs with high confidence values is large in the predicted structures from the accurate multiple alignments. The confidence values of the prediction to the UnaL2 family is plotted in Figure 5.6 (right). The confidence values will be useful to rank the secondary structure candidates in genomic scale studies of conserved secondary structures.

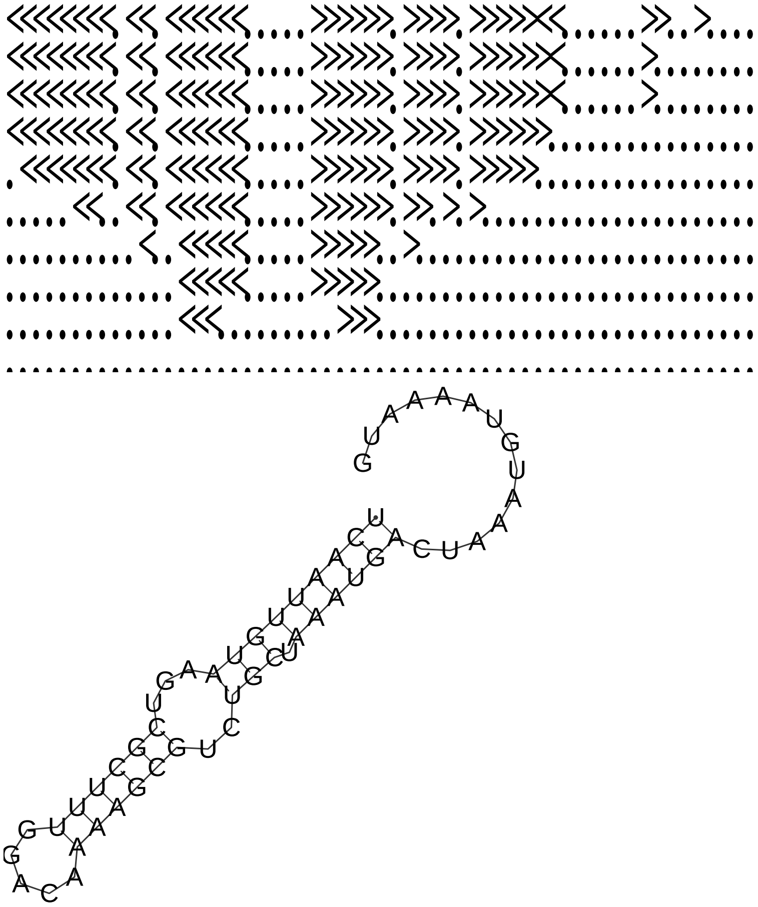


Figure 5.5. An example of the consensus secondary structure prediction for varying the parameters α (left). A ProbCons alignment of the UnaL2 family in Table 5.1 is used. The predictions are made using the McCaskill-MEA algorithm. The corresponding α values are 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.63, 1.26, 2.51 and 5.01 from top to bottom. The true structure is also shown in the figure on the right.

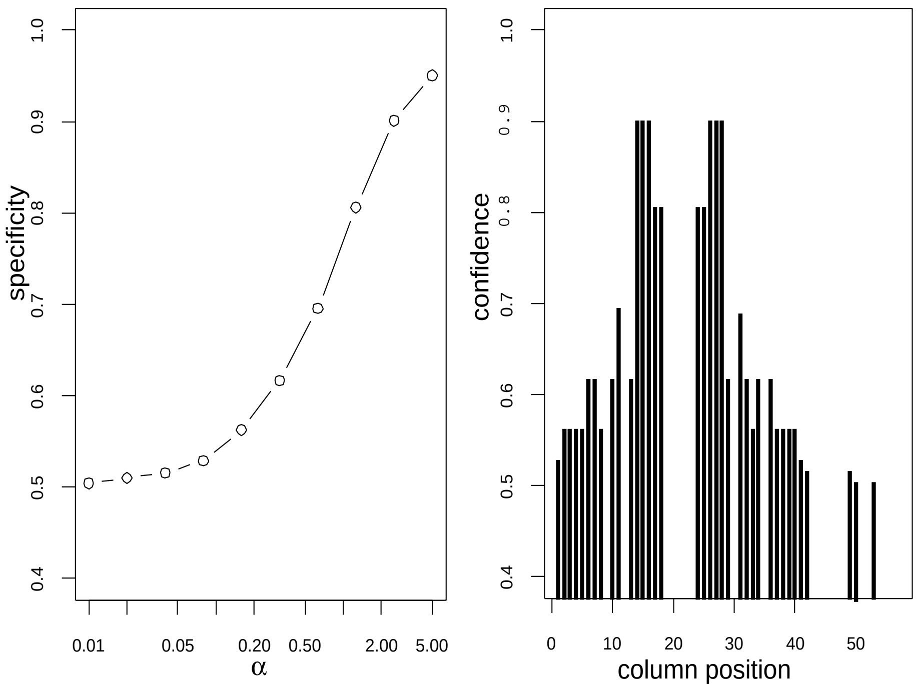


Figure 5.6. Confidence scores of each predicted base pair. In the left-hand-side figure, the relation between the α values and specificity is plotted. The curve is derived from the ROC curve of the McCaskill-MEA predictions from the ProbCons alignments (Figure 5.2). The x axis denotes the α values on a log scale, and the y axis denotes the specificity. In the right-hand-side figure, the confidence values of the predicted base pairs (Figure 5.5) of the UnaL2 family are plotted. The x axis indicates the column position of the alignment, and the y axis indicates the computed confidence values.

5.5. Summary

I have presented a method to predict the conserved secondary structures from multiple aligned sequences that are subject to alignment failures. Our method first calculates the base pairing probability matrix for each sequence, which are subsequently averaged to yield the base pairing probability matrix of the alignment. The consensus secondary structure is obtained by maximizing the expected accuracy of the structure with respect to the base pairing probabilities. For computationally aligned multiple sequences, our method shows a better performance as compared to other frequently used programs. I have shown that our method is particularly suitable for the alignments that suffer from significant alignment failures and that consist of a large number of sequences. I have shown that the parameter α in our model, which controls the sensitivity and specificity of the prediction, is useful for the genomic scale screening of conserved secondary structures and for assigning confidence values to the predicted base pairs.

In the present chapter, I have investigated only the *global* problem of structure prediction, that is, the lengths of the alignments and those of the structural RNA genes are assumed to be of the same order. For the *local* problem of consensus structure prediction that searches long multiple alignments for small conserved structures, the calculation of the base pairing probability matrices over the entire alignment might have problems caused by the stochastic disturbance from the regions that are not related to the RNA genes and secondary structures. I leave the investigation of the local structure prediction problem and the scaling property of the base pairing probability matrices for future study.

I have considered only simple applications of the maximal expected accuracy principle in which the base pairing probabilities are derived either from an alignment or from all the sequences in the alignment. However, I can extend our method to compute the average of both the probabilities that might complement each other. I can also consider combining other probability matrices derived from other models such as SCFG models. Such considerations lead to the problem of finding the best proportionality constants to sum the various probabilities. It may be interesting to study machine-learning approaches to combine various base pairing probabilities in an optimal manner.

Table 5.2. The p values of the Fisher's sign test that tests the null hypothesis that McCaskill-MEA is not more accurate than each of the other programs. The first column shows the name of the program compared. The other columns show the p values that measure the unlikeliness of the null hypothesis. The number of observations that the accuracy of Murlet is better (+) or worse (-) than that of the program being compared is also shown in the round brackets.

Program	Reference	ProbCons	ClustalW
	p-value(+/−)	p-value(+/−)	p-value(+/−)
Pfold	0.67 (39/42)	0.094 (58/16)	7.7e-5 (59/24)
Alifold	0.091 (47/34)	0.0053 (53/29)	0.007 (52/29)

Chapter 6

Conclusion

I have developed an efficient alignment method that accurately aligns multiple structural RNA sequences. The method first computes the base pairing probabilities and the match probabilities. Then a simple Sankoff algorithm is applied to obtain the final alignment using these probabilities. The scoring system of the method enables to incorporate complex secondary structure information and homology information without complicating the Sankoff algorithm. The method have highest accuracy among the existing softwares in both the alignment quality and the accuracy of consensus secondary structure prediction from the alignments. The algorithm has an efficient method to reduce the DP region to be computed which allows alignment of long RNA sequences that have not been computable by other Sankoff-based alignment tools.

I have also proposed a method to predict the conserved secondary structures from multiply aligned sequences which are subject to alignment failures. The method first calculates the base pairing probability matrix for each sequence, which are subsequently averaged over to give the base pairing probability matrix of the alignment. The consensus secondary structure is obtained by maximizing the expected accuracy of secondary structure with respect to the base pairing probabilities. For computationally aligned multiple sequences, the method shows better performance than other frequently used programs. The method is especially suitable for the alignments which suffer from significant alignment failures and consist of a large number of sequences. The parameter $\alpha^{(b)}$ of the model, that controls the sensitivity and specificity of the prediction, is useful for the genomic scale screening of conserved secondary structures, and for assigning confidence values to the predicted base pairs.

Appendix A

Supplements to Chapter 3

A.1. Algorithm for Reducing DP Region

I use the convention that the i-th base of the first sequence that is inserted between $j-1$ and j-th base of the second sequence is emitted at position (i,j) in the DP matrix. In this convention, the size of the DP matrix is $(L^{(1)}+1)\times(L^{(2)}+1)$ for sequences of lengths $L^{(1)}$ and $L^{(2)}$. and the ranges of row and column indices are $1 \leq i \leq (L^{(1)}+1)$ and $1 \leq j \leq (L^{(2)}+1)$ respectively, in order to account for the $3'$-terminal gap insertions. I represent the DP region by two arrays of left and right column boundaries $j_l[i]$ and $j_r[i]$ in the DP matrix. Using these arrays, the DP region is represented by the set $\{(i,j)|1 \leq i \leq (L^{(1)}+1), j_l[i] \leq j \leq j_r[i]\}$.

Algorithm 5 shows the algorithm for reducing the DP region. The initial DP region is represented as $j_l[i]$ and $j_r[i]$. These boundaries are modified to represent the reduced DP region after the computation. The algorithm requires as input the initial DP region $j_l[i]$ and $j_r[i]$, the match probability matrix $p^{(a)}$, the threshold value ϵ and the minimum DP region that enclose the initial DP path, which is represented by $j_{l0}[i]$ and $j_{r0}[i]$.

The reduced DP region has several properties.

- The region is simply connected. In other words, the region has no holes. This is obvious since each slice of the region by rows is represented by only one segment $j_l[i] \leq j \leq j_r[i]$.
- The region includes the initial alignment path $j_l[i] \leq j_{l0}[i] \leq j_{r0}[i] \leq j_r[i]$.

Algorithm 5 Algorithm for reducing the dynamic programming region. $j_l[i]$ and $j_r[i]$ are the left and right column boundaries of the DP region at row i. On input, $j_l[i]$ and $j_r[i]$ represent the strip region around the initial DP alignment path. On output, $j_l[i]$ and $j_r[i]$ represent the reduced DP region. $j_{l0}[i]$ and $j_{r0}[i]$ are the boundaries of the minimum DP region that enclose the initial DP path. $p^{(a)}(i,j)$ is assumed to returns an element of the match probability matrix at position (i,j) if $1 \le i \le L^{(1)}$ and $1 \le j \le L^{(2)}$, and returns 0 otherwise.

Input: $j_l[\cdot]$, $j_r[\cdot]$, $j_{l0}[\cdot]$, $j_{r0}[\cdot]$, ϵ, $p^{(a)}(\cdot,\cdot)$

Output: $j_l[\cdot]$, $j_r[\cdot]$

```
$j_0 \leftarrow 1$
for $i \leftarrow 1 \cdots (L^{(1)} + 1)$ do
    $j_0 \leftarrow \max(j_0, j_{r0}[i])$
    $j \leftarrow j_r[i]$
    $j_r[i] \leftarrow j_0$
    while $j \ge j_0$ do
        if $\epsilon \le p^{(a)}(i,j)$ then
            $j_r[i] \leftarrow j$
            $j_0 \leftarrow (j+1)$
            break
        end if
        $j \leftarrow (j-1)$
    end while
end for
$j_0 \leftarrow L^{(2)}$
for $i \leftarrow (L^{(1)} + 1) \cdots 1$ do
    $j_0 \leftarrow \min(j_0, j_{l0}[i])$
    $j \leftarrow j_l[i]$
    $j_l[i] \leftarrow j_0$
    while $j \le j_0$ do
        if $\epsilon \le p^{(a)}(i-1, j-1)$ then
            $j_l[i] \leftarrow j$
            $j_0 \leftarrow (j-1)$
            break
        end if
        $j \leftarrow (j+1)$
    end while
end for
```

- $j_r[i] \leq j_r[i+1]$ and $j_l[i] \leq j_l[i+1]$.
- for each position (i, j) that has match probability $p^{(a)} > \epsilon$ and is right of the initial path $j > j_{r0}[i]$, the lower left region $\{(i', j')|i' = i, j_{l0}[i'] < j' \leq j\} \cup \{(i', j')|i' > i, j_{r0}[i'] < j' \leq (j+1)\}$ is contained in the reduced DP region.
- for each position (i, j) that has match probability $p^{(a)} > \epsilon$ and is left of the initial path $j < j_{l0}[i]$, the upper right region $\{(i', j')|i' = (i+1), (j+1) \leq j' < j_{l0}[i']\} \cup \{(i', j')|i' \leq i, j \leq j' < j_{l0}[i']\}$ is contained in the reduced DP region.

From the last two properties, it follows that for any position pair (i, j) and (i', j') that have match probabilities $p^{(a)}(i, j), p^{(a)}(i', j') > \epsilon$ and can coexist in an alignment (i.e. $(i < i'$ and $j < j')$ or $(i' < i$ and $j' < j)$), there exists at least one alignment path in the reduced DP region that connect these positions. In fact, the reduced DP region is given by the union of the region corresponding to $j_{l0}[i]$ and $j_{r0}[i]$ and all the upper-left and lower-right regions that are described in the last two properties.

A.2. Novel Accuracy Measures: SQS, SSS, and PCS

To define SQS, SSS and PCS mathematically, I first give a few definitions. Let $\iota_{\mathcal{A}}^{(h)}$ be the mapping from the position $i \in \mathcal{C}^{(h)}$ of sequence $x^{(h)}$ to the corresponding alignment column $I \in \mathcal{C}_{\mathcal{A}}$ in the alignment $\mathcal{A}$

$$\iota_{\mathcal{A}}^{(h)} : \mathcal{C}^{(h)} \longrightarrow \mathcal{C}_{\mathcal{A}}$$
$$h = 1, \cdots, N$$

For each consensus secondary structure $\mathcal{S} = \{\mathcal{L}, \mathcal{P}\}$ of the alignment $\mathcal{A}$, the secondary structure $\mathcal{S}^{(h)}$ of sequence $x^{(h)}$ associated to $\mathcal{S}$ is defined by,

$$\begin{aligned}
\mathcal{S}^{(h)} &= \{\mathcal{L}^{(h)}, \mathcal{P}^{(h)}\} \\
\mathcal{P}^{(h)} &= \{(i, j) \in \mathcal{PC}^{(h)} | \exists (I, J) \in \mathcal{P}, I = \iota^{(h)}(i), J = \iota^{(h)}(j)\} \\
\mathcal{L}^{(h)} &= \{i \in \mathcal{C}^{(h)} | \forall (i', j') \in \mathcal{P}^{(h)}, i \neq i', j'\}
\end{aligned}$$

For each alignment column I in the alignment $\mathcal{A}$, the column vector $c_{\mathcal{A},I}$ is defined as follows,

$$c_{\mathcal{A},I}(h) = \begin{cases} \text{'-' if the column } I \text{ is a gap position for sequence } x^{(h)} \\ \iota_{\mathcal{A}}^{(h)-1}(I) \end{cases}$$
$$h = 1, \cdots, N$$

where $i = \iota_{\mathcal{A}}^{(h)-1}(I)$ is the position of sequence $x^{(h)}$ aligned at the column I.

To compute SQS, the number of quadruples $((i,j),(k,l)) \in \mathcal{P}^{(h)} \times \mathcal{P}^{(h')}$ satisfying the following constraint is computed for each pair $(x^{(h)}, x^{(h')})$ of sequences.

$$\begin{gathered} ((i,j),(k,l)) \in \mathcal{P}^{(h)} \times \mathcal{P}^{(h')} \\ \iota_{\text{ref}}^{(h)}(i) = \iota_{\text{ref}}^{(h)}(k) \\ \iota_{\text{ref}}^{(h)}(j) = \iota_{\text{ref}}^{(h)}(l) \\ \iota_{\text{sbj}}^{(h)}(i) = \iota_{\text{sbj}}^{(h)}(k) \\ \iota_{\text{sbj}}^{(h)}(j) = \iota_{\text{sbj}}^{(h)}(l) \end{gathered}$$

where the subscripts ref and sbj indicate the reference alignment and the subject alignment being evaluated, respectively. Then the count is summed over all the pairs of sequences. The SQS is obtained by taking the ratio of the count of the subject alignment to that of the ideal alignment that is identical to the reference alignment. To compute SSS, the number of quadruples that satisfies the constraint

$$\begin{gathered} ((i,j),(k,l)) \in \mathcal{P}^{(h)} \times \mathcal{P}^{(h')} \\ \iota_{\text{sbj}}^{(h)}(i) = \iota_{\text{sbj}}^{(h)}(k) \\ \iota_{\text{sbj}}^{(h)}(j) = \iota_{\text{sbj}}^{(h)}(l) \end{gathered}$$

is computed. The SSS value is obtained by taking the ratio between the count of the subject alignment and that of the ideal alignment. To compute PCS, the number of pair columns (I, J) that satisfies the constraint

$$\begin{gathered} (I,J) \in \mathcal{PC}_{\text{sbj}} \\ \exists (K,L) \in \mathcal{P}_{\text{ref}} \\ c_{\text{ref},K} = c_{\text{sbj},I} \\ c_{\text{ref},L} = c_{\text{sbj},J} \end{gathered}$$

is calculated. The PCS value is obtained by taking the ratio between the count of the subject alignment and that of the ideal alignment.

Figure A.1 shows examples of the alignments. (a) is the reference alignment, (b) is the subject alignment and the alignment (c) is a copy of the reference alignment used for the comparison. The secondary structures of sequences in the three alignments are derived from the structure annotated to the reference, which are shown in the bottom part of the figure, where the aligned bases are replaced with the corresponding sequence positions. Figure A.2 shows examples of the computation of SQS and SSS values for the multiple alignment of Figure A.1. For the SQS computation, three quadruples $((1,7),(1,9))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$, $((2,8),(1,7))$ and $((3,7),(2,6))$ in $\mathcal{P}^{(2)} \times \mathcal{P}^{(3)}$ contribute to the count for the 'subject' alignment (Figure A.2(a)), while five quadruples $((1,7),(1,9)$ and $((2,6),(2,8))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$, $((2,6),(1,7))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(3)}$, $((2,8),(1,7))$ and $((3,7),(2,6))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(3)}$ contribute to the count for the 'subject0' alignment (Figure A.2(b)). The SQS value is given by the ratio $3/5 = 0.6$. The count that contributes to SQS also contributes to the count for SSS. However the quadruples $((2,6),(3,7))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(2)}$ and $((2,6),(2,6))$ in $\mathcal{P}^{(1)} \times \mathcal{P}^{(3)}$ also contribute to the SSS count (Figure A.2(c)). The count for the 'subject0' alignment is unchanged from that of SQS (Figure A.2(d)). Therefore, SSS is given by $(3+2)/5 = 1$. Figure A.3 shows an example of the PCS computation. Since the pair of column vectors $((1,1,-),(7,9,8))$ exists both in the reference and 'subject' alignments and these columns are annotated to form a base pair in the reference alignment, The pair column $(I,J) = (1,9)$ in $\mathcal{PC}_{\mathrm{sbj}}$ contributes to the count of PCS (Figure A.3(b)). Similarly the three pair columns $(1,9)$, $(2,8)$ and $(3,7)$ in $\mathcal{PC}_{\mathrm{sbj0}}$, whose pair column vectors are $((1,1,-),(7,9,8))$, $((-,2,1),(-,8,7))$, and $((2,3,2),(6,7,6))$, respectively, contribute to the count for the 'subject0' alignment (Figure A.3(c)). The PCS value is then given by the ratio $1/3 \approx 0.33$.

A.3. Consensus Structure Prediction By Stemloc and PMMulti

Table A.1 shows the MCC values of the Pfold predictions to the Stemloc and PMMMulti alignments and the original consensus structure predictions made by

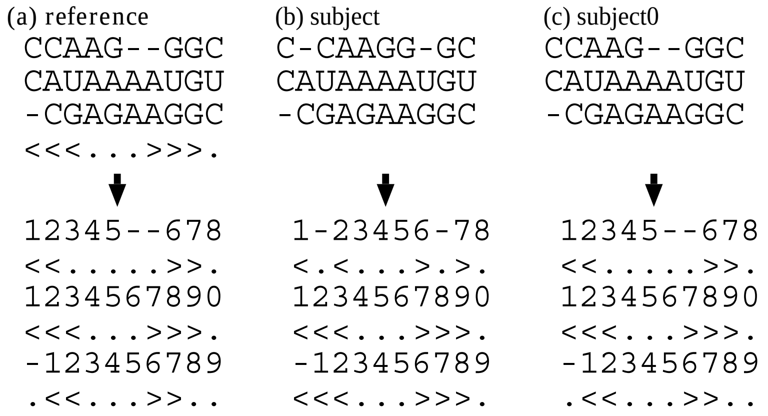


Figure A.1. Derivation of the secondary structures of sequences from the consensus secondary structure of the alignment.

Table A.1. Comparison of MCC values between the predictions made by Pfold and those made by Stemloc and PMMulti. “Average(Stemloc),” “Average(PMMMulti),” and “Average(common)” have the same meanings as those in the main text. “original” indicates the MCC values for the original predictions made by Stemloc and PMMulti.

	Stemloc	PMMulti
	Pfold / original	Pfold / original
Average (Stemloc)	0.67 / 0.58	- / -
Average (PMMulti)	- / -	0.54 / 0.42
Average (common)	0.68 / 0.61	0.54 / 0.42

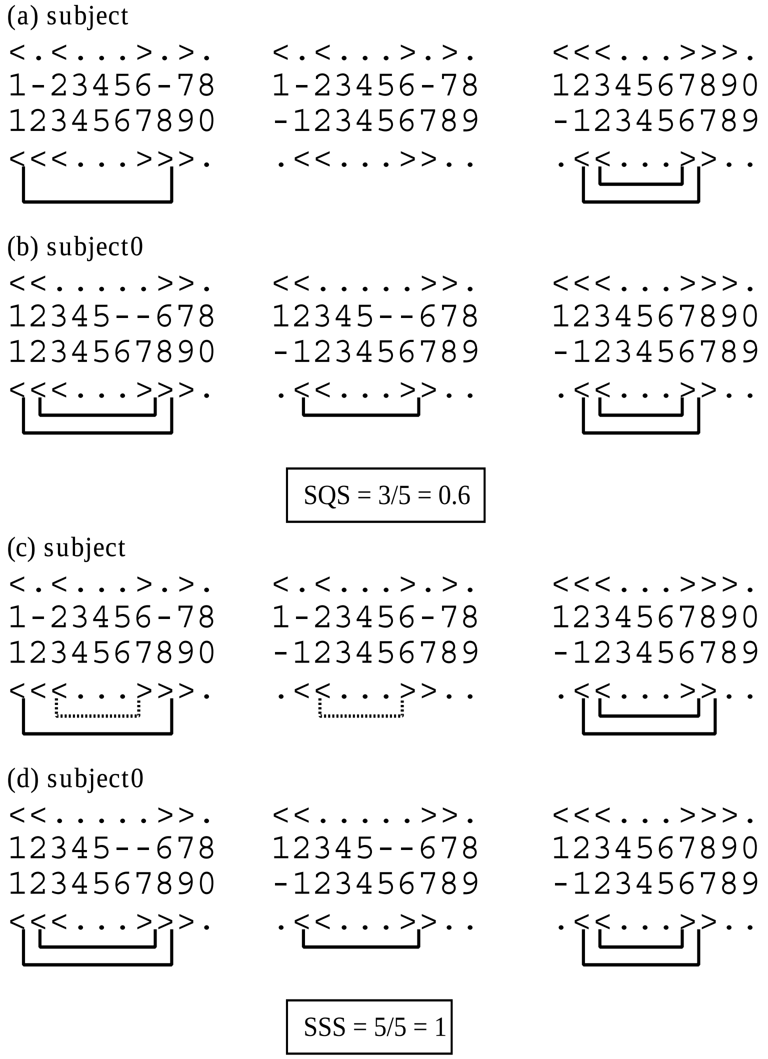


Figure A.2. Examples of the computation of SQS and SSS. The left, center, and right alignments correspond to the sequence pairs $(h, h') = (1, 2), (1, 3),$ and $(2, 3)$, respectively.

these programs. The table shows that the accuracies of the original predictions are almost 10% lower than those of Pfold predictions.

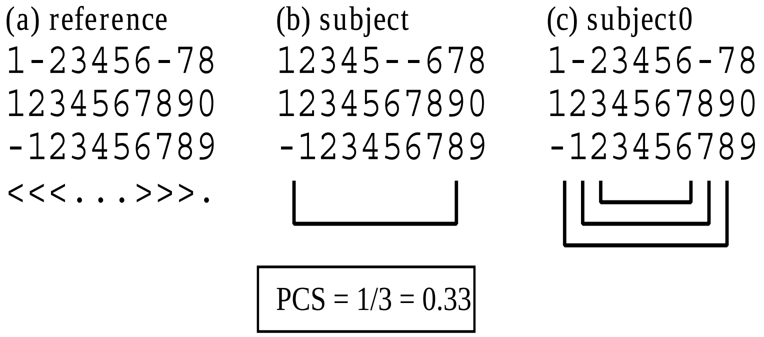


Figure A.3. An example of the computation of PCS.

Appendix B

Test Dataset for Algorithm Comparison

The Rfam accession and ID of the multiple alignments used to compare the algorithms are listed below.

RF00001 5S_rRNA			
U39694.1/9296-9392	M35167.1/2-111	AJ131602.1/3-115	X52302.1/2-117
Y00159.1/2-117	M58387.1/5-112	X67494.1/1-118	M36316.1/2-120
X02706.1/2-120	X05535.1/1-118		
Z50076.1/4-120	X55253.1/3-119	M33888.1/3-119	M34775.1/1-115
X03902.1/1-114	M11546.1/4-118	L37449.1/4-118	X79704.1/3397-3511
X02246.1/3-118	X06834.1/1-119		
U05019.1/544-658	X62858.1/1-121	M11546.1/4-118	L37449.1/4-118
L37450.1/2-118	X00691.1/1-119	X00067.1/1-119	X01004.1/1-119
X52052.1/4-118	M10436.1/2-120		
X62859.1/3-124	M16172.1/3-119	V00646.1/1-107	M34772.1/1-114
X02253.1/3-118	X00475.1/121-239	D10526.1/1-121	X06847.1/1-119
AB015590.1/1-119	X00378.1/2-120		
X14441.1/5-123	M36188.1/5-125	S73542.1/3-119	X12884.1/1-117
X00081.1/2-117	L37446.1/4-117	X02241.1/3-119	K03169.1/1-118
X00998.1/1-119	X52048.1/2-120		
RF00002 5_8S_rRNA			
AF093014.1/662-809	U21939.1/68-218	X54512.1/4422-4569	M14649.1/2015-2169
X53361.2/1206-1368	M63701.1/247-415	U13369.1/6624-6776	AF158724.1/274-427
X80212.2/2104-2256	Y07978.1/48-201		
AF093014.1/662-809	AF223570.1/244-389	AF306774.1/466-634	X99212.1/685-824
AF158724.1/274-427	X00601.1/3997-4154	V01159.1/185-331	X66325.1/2369-2523
Y11511.1/115-268	D16558.1/2710-2862		
AF093014.1/662-809	AF026388.1/2852-3006	U21939.1/68-218	X01533.1/5-152
M63701.1/247-415	M36008.1/959-1112	X00601.1/3997-4154	V01159.1/185-331
AF468917.1/611-763	D10840.1/200-355		
U48228.1/7-166	X53361.2/1206-1368	M63701.1/247-415	X90410.1/15-165
AF196778.1/2-154	L78065.1/3758-3910	M36008.1/959-1112	X03680.1/3159-3308

AF307619.1/287-442	Y07976.1/124-271		

AF093014.1/662-809	X54512.1/4422-4569	U58510.1/2022-2198	X52949.1/857-993
Y00055.1/4327-4494	AL049755.2/30863-31016	AF196778.1/2-154	L78065.1/3758-3910
X03680.1/3159-3308	X00601.1/3997-4154		

RF00061 IRES_HCV

AF046866.1/1-389	D31726.1/1-254	M84834.1/1-321	L34381.1/1-279
AY070180.1/1-277	Z84223.1/1-250	U45462.1/1-333	D13448.1/1-296
U05027.1/1-361	D14307.1/1-327		

AY188170.1/2-196	AF057153.1/1-237	AY190387.1/1-182	AY344028.1/1-165
Z23075.1/1-230	AY190441.1/4-181	AY145993.1/1-216	AY188160.1/2-196
Z84275.1/1-250	U14774.1/1-239		

AY188094.1/3-196	AF041280.1/1-191	Z84228.1/1-250	AY188114.1/3-196
Z84216.1/1-250	AY446063.1/1-272	AY145945.1/1-206	L27900.1/1-231
L29581.1/2-237	AY145980.1/1-251		

Z84240.1/1-250	L34393.1/1-279	Z84248.1/1-250	U23750.1/1-297
D63822.1/1-388	AJ438621.1/1-250	D63857.1/2-298	M84851.1/1-320
AF041278.1/1-191	AF056006.1/1-237		

AF207771.1/1-379	Z36523.1/3-205	U05022.1/1-360	M84828.1/1-321
AJ006323.1/1-251	U14772.1/1-239	U23390.1/1-319	AF506669.1/3-246
L34364.1/1-279	AY163830.1/1-251		

RF00168 Lysine

AE017007.1/287994-288186	AP001512.1/119931-120105	AL596166.1/112469-112272	AE016747.1/182196-182375
AF270308.1/2156-2331	AE007576.1/1562-1747	AE006448.1/6071-6253	AE015829.1/4454-4280
AE006126.1/222-48	AE001799.1/20444-20268		

AE017007.1/287994-288186	AP001512.1/119931-120105	AP004598.1/253855-254037	AP004601.1/22341-22165
AE016747.1/182196-182375	AP004827.1/261938-261763	AE015937.1/285886-286061	AE013149.1/9167-9356
AE015545.1/1265-1436	AP005076.1/290738-290918		

M93419.1/332-511	Z99121.2/6040-5861	AE017028.1/200117-200298	AE017274.1/20257-20449
AL591976.1/186683-186486	AP003362.3/86114-86289	AP004827.1/261938-261763	AE007843.1/1920-1745
AE016947.1/224792-224618	AE010489.1/2647-2468		

Z99121.2/6040-5861	AE017274.1/20257-20449	AP001517.1/215539-215348	AE016747.1/182196-182375
AF270308.1/2156-2331	AP003187.2/139222-139393	AP003194.2/187997-187828	AE013149.1/9167-9356
AP005335.1/123141-123320	AE001799.1/20444-20268		

AE017007.1/287994-288186	AP004827.1/261938-261763	AP003187.2/139222-139393	AE013149.1/9167-9356
AE013039.1/9145-9323	AE016947.1/224792-224618	AE016770.1/235405-235209	AE004193.1/5679-5861
AP005342.1/28132-28310	AE001799.1/20444-20268		

RF00050 RFN

X51510.1/938-1082	AJ010128.1/362-218	AF269712.1/1300-1435	AF269345.1/3001-2863
AE014142.1/1906-2030	AL766851.1/11027-10882	AL766847.1/1-103	AE010459.1/9470-9585
AE004431.1/9358-9220	AE001820.1/156-35		

AP005274.1/66442-66279	AL939108.1/337004-336876	AE014618.1/8796-8645	L09228.1/7992-8136
AP004827.1/34791-34657	AE007574.1/3571-3685	AE006333.1/7352-7468	AP003011.2/168184-168036
AE006176.1/9504-9349	AE004431.1/9358-9220		

AP005214.1/86557-86391	AE014618.1/8796-8645	AE001878.1/30-178	AL591981.1/267609-267487
AL766851.1/11027-10882	AL646083.1/100911-100755	M64472.1/901-1048	U27202.1/210-335
AE006176.1/9504-9349	AE001820.1/156-35		

AP005274.1/66442-66279	AE001878.1/30-178	X95955.1/376-523	AL591981.1/267609-267487
AF269712.1/1300-1435	AE006360.1/976-840	AE014142.1/1906-2030	AE009695.1/6814-6651
AE004431.1/9358-9220	AE012168.1/3092-3260		

AL939108.1/337004-336876	X95955.1/376-523	AJ010128.1/362-218	AP004827.1/34791-34657
AP003187.2/100445-100560	AL591790.1/15823-15987	AL646083.1/100911-100755	M64472.1/901-1048
U27202.1/210-335	AJ009832.1/4549-4670		

RF00175 Retroviral_psi

AF042100.1/691-809	U88826.1/62-176	AB097869.1/693-813	AF110970.1/83-200
AF538302.1/716-833	M17451.1/208-324	AJ291719.1/695-813	AY158534.1/41-159
AY161882.1/12-127	U88822.1/208-327		
K03454.1/240-356	AF286253.1/173-290	AY161881.1/12-129	AF443113.1/69-186
AF076998.1/46-162	AF414006.1/702-819	AF192135.1/86-206	AY161882.1/12-127
AJ288981.1/736-854	AF492624.1/73-186		
AF193253.1/70-186	AB097869.1/693-813	AF164485.1/717-830	AB097865.1/71-189
AF082394.1/14-133	AY162224.1/704-822	AF061640.1/97-212	AF423756.1/144-260
AY173957.1/62-175	AF042105.1/49-165		
K03454.1/240-356	AF190127.1/72-196	AF538304.1/695-811	AF179368.1/10-127
AF110972.1/83-200	AY161885.1/12-128	AB032741.1/666-779	AF286233.1/39-156
AF286236.1/192-308	AY118154.1/240-357		
AJ293865.1/46-165	AF414006.1/702-819	AY161883.1/12-129	AF049494.1/241-357
AY118159.1/240-356	AF443075.1/69-186	AY173957.1/62-175	AF443111.1/69-188
U88822.1/208-327	AF042104.1/28-144		
RF00031 SECIS			
AE003628.2/106178-106240	Y11109.1/1272-1330	L28111.1/1299-1365	AF241527.2/359-424
AF136399.1/1808-1868	AF195142.1/461-524	AF288740.1/1291-1357	AF333036.1/2190-2249
AF093774.1/5851-5916	U43286.1/2054-2120		
AC092237.1/57223-57161	AY119185.1/838-902	Y11111.1/1260-1324	AF125575.1/5781-5843
AL645723.11/192421-192359	AB030643.1/4176-4241	X12367.1/703-764	BC003127.1/865-928
U67171.1/375-442	AF053984.1/1951-2017		
AY060611.1/560-627	Y11109.1/927-987	Y11111.1/1260-1324	AF322071.1/1577-1642
X84742.1/5239-5302	M63574.1/1465-1528	AF166127.1/1919-1981	U43286.1/2054-2120
X76008.1/2709-2772	U61947.2/4246-4309		
AE003628.2/106178-106240	Y11111.1/1260-1324	Y11273.1/1139-1211	X84742.1/5239-5302
AB030643.1/4176-4241	X57999.1/1526-1586	S79854.1/1605-1666	AC000078.2/21139-21077
U67853.1/375-442	U61947.2/4246-4309		
AY060611.1/560-627	AL645723.11/192421-192359	AF241527.2/359-424	AF136399.1/1808-1868
AF288740.1/1291-1357	AC002327.1/156204-156268	BC003127.1/865-928	X53463.1/847-903
AB017534.1/661-726	AF053984.1/1951-2017		
RF00169 SRP_bact			
M31831.1/60-156	AP002546.2/147527-147424	AE012781.1/271-172	D90912.1/112591-112684
AF368293.1/121-224	AE007319.1/10471-10547	X53678.1/99-175	AE009624.1/3419-3517
U32795.1/4504-4603	AE004188.1/3986-4083		
AE000759.1/8335-8411	AP003584.1/338551-338455	D11419.1/119-221	AE006489.1/4483-4560
AE002112.1/7386-7290	AL591782.1/259929-260027	AF203881.1/14955-14875	AL162755.2/146999-146906
AE014122.1/13296-13200	M31830.1/30-129		
AE012781.1/271-172	AP003584.1/338551-338455	AF269814.1/2627-2730	AE007319.1/10471-10547
AE010530.1/10602-10691	U22036.1/386-480	AF203881.1/14955-14875	AF482014.1/12-103
AJ414155.1/123558-123461	AE001187.1/2891-2802		
AE012781.1/271-172	AJ011025.1/3477-3397	AF368293.1/121-224	AE006305.1/11176-11252
AE002112.1/7386-7290	AP003006.2/282610-282707	AE008680.1/732-813	AL139074.2/66646-66744
AE001151.1/4236-4333	AE001187.1/2891-2802		
AL023596.1/21933-22018	M31831.1/60-156	AP002546.2/147527-147424	D90912.1/112591-112684
D11418.1/118-220	U39706.1/5918-5836	AE010530.1/10602-10691	AE003940.1/637-734
AE001151.1/4236-4333	AE001187.1/2891-2802		
RF00017 SRP_euk_arch			
X15364.1/835-1130	AE000940.1/8761-9056	M22560.1/129-422	AE010126.1/9583-9877
AL354512.3/23429-23155	M20837.1/128-397	Z99259.1/7742-7997	Z34533.1/38786-39082
X14661.1/2-308	X65990.1/1-300		
M32222.1/953-1250	U67510.1/7006-7301	M22560.1/129-422	AE013320.1/6338-6044
AE010387.1/4828-4528	AC002512.1/76041-75744	X01055.1/1-297	Z30973.1/7231-6935

Z29104.1/1-303	X65991.1/1-302		
M21085.1/9-299	X17237.1/11-302	AF006750.1/731-1010	M80262.1/106-378
X51658.1/237-504	Z99259.1/7742-7997	AB021174.1/1-299	X13914.1/2-302
X65991.1/1-302	AC005275.1/105500-105803		
AP000058.1/196062-195789	X17239.1/14-300	AF395888.1/197-488	AE000940.1/8761-9056
AE013320.1/6338-6044	X56981.1/102-383	X01055.1/1-297	AP003253.3/106424-106740
Z29099.1/1-303	AB020752.1/35792-36096		
X17239.1/14-300	X15364.1/835-1130	AF006750.1/731-1010	M80262.1/106-378
X51658.1/237-504	AC002512.1/76041-75744	X01037.1/5-303	Z30973.1/7231-6935
X65990.1/1-300	AC005275.1/105500-105804		
RF00162 S_box			
Z99111.2/15242-15342	AP001518.1/300422-300310	AP001511.1/149016-149129	AP001518.1/137068-136956
AP004603.1/162173-162282	AL596170.1/192-86	AL596172.1/176254-176136	AL596163.1/172398-172500
AP004829.1/296359-296264	AE007608.1/12720-12617		
Z99123.2/188544-188650	Z99109.2/169069-168952	AP001511.1/149016-149129	AP004596.1/197999-197893
AP004603.1/200738-200630	AL591976.1/242770-242867	AL591983.1/96176-96058	AL591980.1/66651-66543
AF269983.1/571-671	AE007614.1/6196-6300		
Z99110.2/47875-47995	AP004601.1/38307-38208	AP004596.1/197999-197893	AP004604.1/167020-166901
AL596170.1/192-86	AL591983.1/96176-96058	AL596166.1/166748-166844	AL591974.1/109385-109266
AE016744.1/15971-16071	AP004828.1/56121-56010		
AJ000974.1/281-386	AF027868.1/5245-5154	Z99123.2/187154-187260	Z99109.2/169069-168952
AP001518.1/300422-300310	AP001518.1/137068-136956	AL591976.1/242770-242867	AF269983.1/571-671
AP004828.1/56121-56010	AP003193.2/95783-95678		
AE012834.1/3852-3958	Z99123.2/188544-188650	AP001511.1/149016-149129	AP004602.1/9845-9739
AP004600.1/266413-266303	AL596163.1/172398-172500	AF269983.1/571-671	U36379.1/1-106
AP004829.1/296359-296264	AE015940.1/18048-17945		
RF00230 T-box			
AE017001.1/248233-248002	AE017002.1/188131-187860	AE017033.1/112246-111996	AF188935.1/65129-65365
AE017038.1/120994-120749	AE017002.1/145236-145479	AE017012.1/289721-289478	AL596169.1/42587-42347
AL596169.1/50907-50661	AP003362.3/336155-335957		
AE017038.1/176480-176232	AE017002.1/188131-187860	AE017033.1/112246-111996	AE017012.1/276630-276387
AE017028.1/41743-41995	AE017028.1/179270-179514	AE017028.1/42023-42282	AL596169.1/7629-7387
AE016948.1/260670-260899	AE016948.1/46912-47109		
AE017002.1/174913-175177	AE017029.1/241355-241630	AE017012.1/106690-106445	AE017028.1/179270-179514
AE017007.1/181606-181356	AL596169.1/7629-7387	AL596169.1/134070-133812	AL591980.1/61532-61313
AE016948.1/260670-260899	AE008470.1/7251-7489		
AE017038.1/176480-176232	AE017012.1/106690-106445	AE017028.1/41743-41995	AE017007.1/181315-181066
AL591980.1/26893-26637	AL596169.1/50907-50661	AL591980.1/61532-61313	AE016747.1/155449-155637
AP004826.1/178432-178626	AE007811.1/6864-6593		
AE017001.1/248233-248002	AE017028.1/222027-221756	AE017038.1/176480-176232	AE017012.1/276630-276387
AE017028.1/41743-41995	AE017002.1/145236-145479	AE017007.1/181315-181066	AE017001.1/297992-298250
AL596169.1/99976-99747	AL596169.1/134070-133812		
RF00059 THI			
AP005276.1/73231-73124	AP005217.1/259663-259772	AP005280.2/150221-150333	AE017179.1/109208-109098
Z82044.1/19193-19300	AP004825.1/152111-152011	AL646057.1/132458-132357	AE016763.1/55154-55058
AE016785.1/288775-288889	AE011282.1/7576-7478		
AP005220.1/194416-194530	AD000014.1/41361-41472	AE016928.1/199416-199321	D64004.1/130312-130407
AP004594.1/205048-204933	AE007789.1/9122-9015	AP002998.2/95169-95056	AL162753.2/24984-25083
AL139075.2/104313-104417	AE012556.1/212138-212040		
Z82044.1/19193-19300	AE017040.1/281858-281743	AE017026.1/171648-171751	AP004595.1/78821-78718
AP004595.1/261047-260946	AL596164.1/123476-123581	AE013167.1/14408-14306	AL646057.1/132458-132357
AE016836.1/195900-195996	AP005345.1/54245-54147		

AY102616.1/4667-4777	BX248356.1/234808-234920	AE017178.1/187260-187147	AE001862.1/178389-178274
U64312.1/2271-2374	AE016999.1/106602-106704	AE013127.1/3291-3189	BX571860.1/177712-177587
AE013968.1/546-637	AF279106.2/41735-41832		
AE014730.1/8718-8610	AE007806.1/8808-8702	AE005876.1/3727-3622	AE017257.1/202839-202737
AE001547.1/1336-1446	AE015414.1/5697-5597	AE006165.1/9171-9068	AE011282.1/7576-7478
AK119882.1/1-98	AK120238.1/2075-2184		
RF00003 U1			
Y00131.1/944-1108	X01749.1/448-581	L22246.1/3195-3357	AC006665.1/5230-5065
X13842.1/1-152	X70869.1/1-161	X06880.1/1-162	X69328.1/1-158
Z11882.1/353-508	X15927.1/1-160		
X63783.1/598-755	X63783.1/1396-1555	AC004546.1/16021-16184	X01749.1/448-581
L22246.1/5536-5699	AC006665.1/5230-5065	M73768.1/361-517	X13841.1/397-552
X69331.1/1-139	X14414.1/152-312		
X63783.1/2026-2185	X01725.1/69-232	M59827.1/771-934	X75936.1/1034-1194
X04994.1/647-780	X13842.1/1-152	X70869.1/1-161	X69333.1/1-157
X15928.1/1-160	AB023028.1/448-285		
X02585.1/923-1086	AL137798.8/45314-45477	X69332.1/1-142	X69328.1/1-158
X69331.1/1-139	X69334.1/1-160	Z11883.1/361-516	X14417.1/177-340
X14414.1/152-312	AB023028.1/448-285		
X55773.1/387-534	AE003745.3/26197-26359	X01091.1/442-605	X75936.1/1034-1194
L22246.1/5536-5699	X70869.1/1-161	X06880.1/1-162	X69331.1/1-139
Z11882.1/353-508	X14419.1/178-338		
RF00004 U2			
AY007785.1/849-1041	AY007788.1/537-679	AY205287.1/148-4	X55772.1/223-412
AF053589.1/90-279	X04241.1/85-276	X51378.1/335-525	L25918.1/1-181
X15930.1/1-195	X06473.1/389-584		
X63786.1/549-738	M23361.1/1-186	X04241.1/85-276	X00093.1/360-550
M12856.1/361-551	X51379.1/254-444	X51372.1/210-400	S72337.1/1-193
M72888.1/1-195	Z37973.1/1-173		
X63784.1/412-602	X63786.1/549-738	M58665.1/571-712	X04244.1/85-276
K02457.1/1-187	M12856.1/361-551	X54113.1/230-415	L22247.1/6321-6513
X51375.1/389-580	X56322.1/513-709		
AF326335.1/1-142	X56457.1/243-390	M14625.1/332-488	M23361.1/1-186
X04244.1/85-276	AF287992.1/4918-5108	X54113.1/230-415	X51374.1/284-474
X51375.1/389-580	Z37972.1/1-174		
X63786.1/1152-1341	X56456.1/243-390	AY205287.1/148-4	X55772.1/223-412
X04256.1/85-275	X00093.1/360-550	X51379.1/254-444	X71483.1/1-191
Z37972.1/1-174	X06475.1/589-783		
RF00436 UnaL2			
AC145764.2/2295-2245	AC144487.1/66140-66085	AL732411.14/41691-41637	AL954371.6/110643-110589
AL953899.7/42720-42666	AL928824.13/119883-119934	BX088699.4/105142-105196	AL928701.7/182087-182139
BX569783.4/61302-61355	AL935128.13/109621-109676		
AC145510.2/144559-144615	AC145764.2/6538-6588	AL929391.10/13836-13883	AL928701.7/176380-176432
BX664748.7/7629-7575	BX569783.4/77351-77301	AL928834.15/61498-61552	BX004821.5/36020-35966
BX649566.3/34014-33960	AL928834.15/4130-4184		
AB029447.1/1210-1265	AC145764.2/53928-53984	BX890619.6/6528-6474	AL954371.6/99505-99453
BX248111.7/42817-42871	BX005334.9/36002-35945	BX569783.4/77351-77301	BX004821.5/46607-46553
BX005301.9/81584-81530	AL928834.15/4130-4184		
AC145764.2/2295-2245	AC146543.2/7696-7640	AC145510.2/122254-122304	AB001842.1/105-159
AL954371.6/55095-55041	BX537162.5/125134-125080	BX005334.9/52316-52370	AL928701.7/19905-19959
AC139110.4/26593-26647	AL935128.13/109621-109676		
AB001858.1/305-359	AL591676.10/16205-16259	AL807818.14/154315-154373	AL928834.15/187161-187215
AL935128.13/73646-73700	BX664748.7/7629-7575	AL592062.11/88031-88081	BX088699.4/24632-24686

BX293564.5/115366-115419	AF397467.1/9309-9363		
RF00181 sno_14q_I_II			
AC121784.2/51879-51950	AC121784.2/52659-52730	AC121784.2/69447-69528	AB014878.1/886-973
AB014878.1/1781-1860	AL132709.5/136822-136751	AL132709.5/149496-149422	AL132709.5/129680-129609
AL132709.5/182043-181969	AL132709.5/167225-167151		
AC121784.2/82332-82403	AC121784.2/48567-48638	AC121784.2/52659-52730	AB014879.1/10-96
AL132709.5/136822-136751	AL132709.5/129680-129609	AL132709.5/145774-145703	AL132709.5/178148-178075
AL132709.5/145122-145048	AL132709.5/148004-147935		
AC121784.2/82332-82403	AB014883.1/1284-1367	AL132709.5/175950-175879	AL132709.5/136822-136751
AL132709.5/135542-135471	AL132709.5/129680-129609	AL132709.5/180473-180397	AL132709.5/142597-142509
AL132709.5/156819-156747	AL132709.5/128363-128289		
AC121784.2/46958-47030	AC121784.2/48567-48638	AC121784.2/77918-77991	AB014878.1/886-973
AB014883.1/1284-1367	AL132709.5/182043-181969	AL132709.5/145122-145048	AL132709.5/158545-158469
AL132709.5/133438-133369	AL132709.5/169743-169666		
AC121784.2/75479-75547	AC121784.2/46958-47030	AC121784.2/51879-51950	AC121784.2/78528-78602
AB076245.1/105-176	AC121784.2/77918-77991	AL132709.5/175950-175879	AL132709.5/134553-134482
AL132709.5/128363-128289	AL132709.5/169743-169666		
RF00005 tRNA			
M26977.1/379-453	AF105125.1/104-176	Z11874.1/40212-40285	Z11880.1/281-353
X14822.1/1-73	K02456.1/141-212	V00654.1/12038-12108	X17318.1/109-39
AJ243756.1/1-71	X67736.1/4837-4923		
X16748.1/1-73	AF041468.1/43811-43739	J01390.1/12028-12098	X05226.1/35-116
J01373.1/73-144	X55342.1/30-101	M19493.1/263-336	X03602.1/660-731
X14848.1/2654-2728	AB017063.1/58819-58900		
J01390.1/6861-6932	J05395.1/2325-2252	K00228.1/1-82	AC009395.7/99012-98941
J04815.1/3159-3231	M20972.1/1-72	M68929.1/151018-150946	X00360.1/1-73
X12857.1/421-494	M16863.1/21-94		
U18089.1/221-293	J01390.1/6449-6519	X66594.1/101-182	M10217.1/9797-9871
X52392.1/5025-5096	X52392.1/6573-6508	J01435.1/6776-6846	AL590385.23/26129-26058
AJ400848.1/29803-29731	X13558.1/186-115		
M16450.1/142-214	J01390.1/6449-6519	X02173.1/54-135	AF200843.1/3014-3079
X14848.1/3824-3891	J01435.1/6776-6846	AC067849.6/4771-4840	AF134583.1/1599-1666
U25144.1/1062-991	X07377.1/52-124		

Appendix C

Test Dataset from Hammerhead Ribozyme Family

The Rfam accession and ID of the multiple alignments used in Figures 3.4 and 5.4 are listed below.

RF00008 Hammerhead_3

AJ295015.1/58-1
AJ536620.1/206-152
AJ005303.1/56-3
AJ241841.1/57-3
AJ295018.1/58-1
AJ295015.1/58-1
AF170504.1/284-337
AJ005303.1/56-3
AJ536615.1/1-44
AJ536614.1/206-152
AJ536612.1/206-152
AJ536620.1/206-152
AJ536620.1/206-152
AJ536617.1/1-40
AJ536619.1/206-152
AJ241840.1/56-3
AJ550911.1/56-3
AJ005305.1/56-3
AJ295015.1/58-1
AJ536620.1/1-40
AJ536614.1/206-152
AJ295015.1/58-1
AJ295018.1/58-1
AJ005302.1/281-334
AJ536620.1/206-152
AJ536620.1/1-40
AJ247121.1/133-53
AF170504.1/284-337
AJ536620.1/206-152
AJ247113.1/134-53
AJ536619.1/206-152
AJ295015.1/58-1
AJ536614.1/206-152
AJ005321.1/281-333
AJ536619.1/206-152

AJ247122.1/132-52
AJ241841.1/57-3
AJ241838.1/56-3
AF170516.1/283-335
AJ536620.1/206-152
AF339740.1/56-3
AJ241833.1/282-334
AJ550909.1/56-3
AJ536612.1/206-152
Y14700.1/133-53
AJ005298.1/56-3
AF170520.1/282-335
AJ550901.1/282-334
AJ536619.1/206-152
AF170503.1/55-3
AF170509.1/56-3
AJ247113.1/134-53
AJ550908.1/281-334
AF170519.1/55-3
AJ241840.1/56-3
AJ247123.1/132-52
AF170503.1/280-333
AJ550912.1/56-3
AJ005319.1/56-3
AF170503.1/55-3
AJ005310.1/56-3
AJ005322.1/281-334
AJ005303.1/56-3
AJ005298.1/56-3
AJ550903.1/281-333
AF170499.1/56-3
AJ536619.1/206-152
AJ005298.1/56-3
AJ550908.1/281-334
AJ241847.1/281-334

AJ550911.1/282-335
AJ005322.1/56-3
AJ550907.1/281-333
AJ005319.1/56-3
AJ241843.1/56-3
AJ550909.1/56-3
AF170519.1/55-3
AJ550909.1/282-333
AF170503.1/55-3
AF170516.1/283-335
AJ241845.1/282-335
AJ550900.1/56-3
AJ241847.1/281-334
AJ247122.1/132-52
AJ005318.1/56-3
AF170499.1/56-3
AJ550906.1/56-3
AJ550898.1/282-335
AJ550907.1/281-333
M33000.1/55-110
M83545.1/282-335
AJ005302.1/281-334
AJ005302.1/281-334
AJ005294.1/282-334
AJ550909.1/282-333
AJ247121.1/133-53
Y12833.1/339-285
AJ005302.1/281-334
AJ241843.1/56-3
AJ550899.1/56-3
AJ247123.1/132-52
AJ241828.1/56-3
AJ241847.1/281-334
AF339739.1/56-3
AJ005322.1/281-334

M33000.1/55-110
AJ241823.1/282-335
M33001.1/56-111
Y12833.1/339-285
J02439.1/42-95
AJ241831.1/281-334
J02386.1/42-95
J02386.1/42-95
AJ005299.1/282-335
Y12833.1/339-285
AJ247116.1/133-53
Y12833.1/339-285
AF170519.1/55-3
AJ005294.1/282-334
AJ550898.1/282-335
M33001.1/56-111
D00685.1/1-46
M33001.1/56-111
AJ005294.1/282-334
M17439.1/1-48
M33000.1/55-110
AJ550909.1/282-333
AJ005322.1/281-334
J02386.1/42-95
AJ005314.1/281-334
AJ241831.1/281-334
M33001.1/56-111
J02439.1/42-95
AJ550909.1/282-333
M33000.1/55-110
J02386.1/42-95
Y14700.1/133-53
AJ550909.1/282-333
J02386.1/42-95
J02439.1/42-95

AJ536614.1/206-152
AF170503.1/280-333
AJ550908.1/281-334
AJ295018.1/58-1
AJ536614.1/206-152
AJ536617.1/1-40
AJ536619.1/206-152
AJ550901.1/282-334
AJ005302.1/281-334
AJ536615.1/1-44
AJ295018.1/58-1
AJ550909.1/56-3
AJ295018.1/58-1
AJ241833.1/282-334
AJ536612.1/206-152
AF170504.1/284-337
AJ550901.1/282-334
AJ536620.1/206-152
AF170503.1/280-333
AJ295018.1/58-1
AJ536612.1/206-152
AJ536617.1/1-40
AJ536617.1/1-40
AJ536619.1/206-152
AF170503.1/280-333
AJ247113.1/134-53
AJ005312.1/56-3
AJ536612.1/206-152
AJ550906.1/56-3
AJ005321.1/281-333
AJ241843.1/56-3
AJ536617.1/1-40
AJ536612.1/206-152
AJ536619.1/206-152
AJ550906.1/56-3
AJ295015.1/58-1
AJ295018.1/58-1
AJ295015.1/58-1
AJ536620.1/1-40
AJ536615.1/1-44
AJ241819.1/56-3
AJ536620.1/1-40
AJ536617.1/1-40
AJ295018.1/58-1
AJ536620.1/206-152
AJ295018.1/58-1
AJ295015.1/58-1
AJ550901.1/282-334
AJ536612.1/206-152
AJ550898.1/282-335
AJ295018.1/58-1
AJ295015.1/58-1
AJ536614.1/206-152
AJ295018.1/58-1
AJ295015.1/58-1
M83545.1/56-3
AJ550909.1/56-3
AJ241819.1/56-3
AF170503.1/55-3
AJ295015.1/58-1
AJ241828.1/56-3
AJ536620.1/1-40
AJ536615.1/1-44
AJ536614.1/206-152
AJ550906.1/56-3
AJ295015.1/58-1
AJ241819.1/56-3
AJ536620.1/206-152
AJ247113.1/134-53
AJ241839.1/282-334
AF170503.1/55-3
AJ005312.1/282-335
AJ005305.1/56-3
AF170509.1/56-3
AJ536619.1/206-152
AJ550912.1/56-3
AJ241840.1/56-3
AJ241850.1/282-334
AF339740.1/56-3
AJ005318.1/56-3
AJ241850.1/282-334
AJ550906.1/282-334
AF339740.1/56-3
AJ247121.1/133-53
AJ550910.1/282-336
AJ005310.1/56-3
AF170523.1/55-3
AJ005312.1/56-3
AJ005321.1/281-333
AJ241843.1/56-3
AJ241839.1/282-334
AJ550903.1/281-333
AJ005312.1/56-3
AF170523.1/55-3
AJ241839.1/282-334
AJ247122.1/132-52
AJ005303.1/56-3
AF170509.1/56-3
AF170509.1/56-3
AJ241850.1/282-334
AJ536614.1/206-152
AJ550909.1/56-3
AF170499.1/56-3
AJ550899.1/56-3
AF170504.1/284-337
AJ005322.1/56-3
AJ536614.1/206-152
AJ241828.1/56-3
AJ550901.1/282-334
AJ241845.1/282-335
Y14700.1/133-53
AF170509.1/56-3
AJ005321.1/281-333
AJ536615.1/1-44
AJ536615.1/1-44
AJ536617.1/1-40
AF170519.1/55-3
AJ241833.1/282-334
AF170516.1/283-335
AJ241839.1/282-334
AJ550903.1/281-333
AJ005294.1/282-334
AJ241828.1/56-3
AJ005305.1/56-3
AJ005300.1/282-335
AJ550898.1/282-335
AJ241843.1/56-3
AJ241845.1/282-335
AJ241833.1/282-334
AJ005305.1/56-3
AJ536619.1/206-152
AJ241845.1/282-335
AJ241831.1/56-3
M83545.1/282-335
AJ536614.1/206-152
AJ247121.1/133-53
AJ241840.1/56-3
AJ241833.1/282-334
AJ241819.1/56-3
AJ550910.1/282-336
J02386.1/42-95
AF170509.1/56-3
AF170516.1/283-335
AJ550911.1/56-3
AJ247113.1/134-53
Y14700.1/133-53
AF170520.1/282-335
J02386.1/42-95
AJ247116.1/133-53
AJ005314.1/281-334
AJ550911.1/282-335
AJ241828.1/56-3
AJ241831.1/281-334
J02439.1/42-95
AF170520.1/282-335
AJ005314.1/281-334
AJ005294.1/282-334
M83545.1/282-335
AJ005322.1/281-334
AJ550906.1/282-334
AJ005322.1/56-3
AJ550906.1/282-334
AJ005299.1/282-335
AJ005305.1/56-3
M33000.1/55-110
M83545.1/282-335
AJ005314.1/281-334
AJ550899.1/56-3
AJ550910.1/282-336
AJ536619.1/206-152
AJ005312.1/282-335
AF170520.1/282-335
AJ005314.1/281-334
AJ241840.1/56-3
AJ005300.1/282-335
AJ005302.1/281-334
AJ241823.1/282-335
AJ247121.1/133-53
AJ550909.1/282-333
AJ550900.1/56-3
AJ241850.1/282-334
AJ005305.1/56-3
AJ005322.1/56-3
AJ550900.1/56-3
AJ005322.1/281-334
AJ241831.1/281-334
AF339739.1/56-3
M83545.1/282-335
AJ550901.1/282-334
AJ247122.1/132-52
M63666.1/246-192
AJ241847.1/281-334
AJ241847.1/281-334
AJ005319.1/56-3
Y12833.1/339-285
AJ550911.1/282-335
J02439.1/42-95
AJ005321.1/281-333
AJ005312.1/282-335
AJ241847.1/281-334
AJ247122.1/132-52
AJ550907.1/281-333
J02386.1/42-95
AJ241828.1/56-3
M63666.1/246-192
M63666.1/246-192
AJ005318.1/56-3
AJ247121.1/133-53
Y12833.1/339-285
M17439.1/1-48
AJ005322.1/281-334
AJ550907.1/281-333
AJ550908.1/281-334
J02386.1/42-95
M33001.1/56-111
M33001.1/56-111
M33001.1/56-111
D00685.1/1-46
J02439.1/42-95
M63666.1/246-192
M33000.1/55-110
J02439.1/42-95
M33001.1/56-111
J02386.1/42-95
D00685.1/1-46
M33000.1/55-110
AJ550910.1/282-336
M33001.1/56-111
J02439.1/42-95
AJ247116.1/133-53
AJ247121.1/133-53
J02386.1/42-95
Y12833.1/339-285
D00685.1/1-46
M33000.1/55-110
Y12833.1/339-285
M63666.1/246-192
J02386.1/42-95
M17439.1/1-48
J02386.1/42-95
AJ550909.1/282-333
M33000.1/55-110
AJ005300.1/282-335
M33000.1/55-110
AJ005314.1/281-334
D00685.1/1-46
Y12833.1/339-285
J02439.1/42-95
M83545.1/282-335
D00685.1/1-46
AJ005314.1/281-334
AJ550907.1/281-333
J02386.1/42-95
J02439.1/42-95
J02439.1/42-95
AJ550907.1/281-333
M33001.1/56-111
AF170499.1/56-3
M17439.1/1-48
J02386.1/42-95
AJ005300.1/282-335
AJ005299.1/282-335
M33000.1/55-110
M33001.1/56-111
M33001.1/56-111
D00685.1/1-46
AJ005302.1/281-334
J02439.1/42-95
M63666.1/246-192
AF170499.1/56-3
J02439.1/42-95
M33001.1/56-111
AJ005321.1/281-333
D00685.1/1-46
M33001.1/56-111
M33001.1/56-111

AJ536614.1/206-152
AJ550908.1/281-334
AJ005320.1/281-333
AJ536614.1/206-152
AJ536614.1/206-152
AJ241839.1/282-334
AJ241839.1/282-334
AJ241831.1/56-3
AJ536619.1/206-152
AJ536620.1/206-152
AJ550908.1/281-334
AJ295015.1/58-1
AJ536620.1/1-40
AJ295018.1/58-1
AJ295018.1/58-1
AJ295018.1/58-1
AJ536619.1/206-152
AJ241840.1/56-3
AJ295018.1/58-1
AF339740.1/56-3
AJ295015.1/58-1
AJ005320.1/281-333
AJ241839.1/282-334
AJ295018.1/58-1
AJ550909.1/56-3
AJ536612.1/206-152
AJ536612.1/206-152
AJ295015.1/58-1
AJ536619.1/206-152
AJ241828.1/56-3
AJ536615.1/1-44
AJ295015.1/58-1
AJ536617.1/1-40
AJ295015.1/58-1
AJ550906.1/56-3
AJ295015.1/58-1
AF170503.1/280-333
AJ550901.1/282-334
AJ005305.1/56-3
AJ536620.1/206-152
AJ550903.1/281-333
AJ536619.1/206-152
AJ536615.1/1-44
AJ241841.1/57-3
AJ536617.1/1-40
AF339740.1/56-3
AJ295015.1/58-1
AF170504.1/284-337
AJ536619.1/206-152
AF170503.1/55-3
AF170504.1/284-337
AJ536612.1/206-152
AJ536620.1/1-40
AJ536620.1/206-152
AJ295015.1/58-1
AJ005298.1/56-3
AJ295018.1/58-1
AJ550903.1/281-333
AJ536612.1/206-152
AF170503.1/55-3
AJ536612.1/206-152
AF170503.1/280-333
AJ536612.1/206-152
AJ247113.1/134-53
AJ536612.1/206-152
AJ536620.1/206-152
AJ295015.1/58-1
AJ536612.1/206-152
AJ536612.1/206-152

AJ005320.1/281-333
AJ005322.1/281-334
AJ005303.1/56-3
AJ550898.1/282-335
AF170520.1/282-335
AJ241843.1/56-3
AJ005312.1/282-335
AJ247121.1/133-53
AJ550901.1/282-334
AJ241831.1/56-3
AJ247116.1/133-53
AF339739.1/56-3
AJ247113.1/134-53
AJ241840.1/56-3
AF170503.1/55-3
AJ241833.1/282-334
Y14700.1/133-53
AF170523.1/55-3
AJ536614.1/206-152
AJ550907.1/281-333
AJ536619.1/206-152
AF339739.1/56-3
AF170523.1/55-3
AJ536617.1/1-40
AJ005299.1/282-335
AF170504.1/284-337
AJ241839.1/282-334
AJ536620.1/1-40
AJ005310.1/56-3
AJ005294.1/282-334
AJ005298.1/56-3
AJ005321.1/281-333
AJ005303.1/56-3
AF170503.1/280-333
Y14700.1/133-53
AJ550901.1/282-334
AJ005302.1/281-334
AJ005303.1/56-3
AJ550909.1/282-333
AJ005303.1/56-3
AF170503.1/55-3
AJ550911.1/282-335
AJ550908.1/281-334
AF170520.1/282-335
AJ005321.1/281-333
AJ550911.1/282-335
AJ241833.1/282-334
J02386.1/42-95
AJ247113.1/134-53
AJ241823.1/282-335
AJ005318.1/56-3
AF339740.1/56-3
AF339740.1/56-3
AJ241828.1/56-3
AF170503.1/55-3
AF170503.1/55-3
AJ536612.1/206-152
AF339739.1/56-3
AJ550901.1/282-334
AJ005294.1/282-334
AF339740.1/56-3
AJ550906.1/56-3
AJ247113.1/134-53
AF170509.1/56-3
Y14700.1/133-53
AF170503.1/280-333
AJ005298.1/56-3
AJ536617.1/1-40
AJ005322.1/56-3

AJ550907.1/56-3
AJ550910.1/282-336
AJ247123.1/132-52
M33001.1/56-111
AJ550900.1/56-3
AF170520.1/282-335
AJ550909.1/282-333
AJ005312.1/282-335
AJ005320.1/281-333
AF170516.1/283-335
AJ550900.1/56-3
AJ550909.1/282-333
AJ005319.1/56-3
AJ550899.1/56-3
AJ550908.1/281-334
AF339740.1/56-3
AF339739.1/56-3
AJ241847.1/281-334
AJ005312.1/56-3
Y12833.1/339-285
AJ550907.1/56-3
AJ247121.1/133-53
AJ550899.1/56-3
AJ247113.1/134-53
AJ550909.1/282-333
AJ241823.1/282-335
AJ247116.1/133-53
AF170503.1/280-333
AJ241830.1/282-334
J02386.1/42-95
AF170516.1/283-335
AJ241838.1/56-3
AJ550911.1/282-335
AJ005318.1/56-3
AJ550898.1/282-335
AJ550907.1/56-3
J02439.1/42-95
AF170509.1/56-3
Y12833.1/339-285
AJ241850.1/282-334
AJ247123.1/132-52
AJ550900.1/56-3
AJ005318.1/56-3
M33000.1/55-110
AF170509.1/56-3
M63666.1/246-192
AJ241841.1/57-3
M33000.1/55-110
AF170503.1/55-3
AF170499.1/56-3
AJ005312.1/282-335
Y12833.1/339-285
AF170503.1/55-3
AF170509.1/56-3
AF339739.1/56-3
AJ005318.1/56-3
Y14700.1/133-53
AJ241838.1/56-3
AJ241819.1/56-3
AJ550900.1/56-3
AJ550909.1/56-3
AJ550907.1/281-333
AJ241840.1/56-3
AJ550911.1/282-335
AJ005314.1/281-334
AJ005319.1/56-3
AJ005320.1/281-333
M83545.1/282-335
AJ005294.1/282-334

AJ550910.1/282-336
M33000.1/55-110
M33001.1/56-111
D00685.1/1-46
M83545.1/282-335
J02439.1/42-95
M33001.1/56-111
M63666.1/246-192
AJ550899.1/56-3
AJ005299.1/282-335
M63666.1/246-192
M33001.1/56-111
AJ005312.1/282-335
M83545.1/282-335
AJ550909.1/282-333
AJ550898.1/282-335
AJ005294.1/282-334
M33001.1/56-111
AF170520.1/282-335
J02439.1/42-95
AJ550909.1/282-333
M33000.1/55-110
J02386.1/42-95
AJ550910.1/282-336
M33000.1/55-110
AF170519.1/55-3
M17439.1/1-48
AF339739.1/56-3
M33001.1/56-111
M17439.1/1-48
M63666.1/246-192
AJ550899.1/56-3
AJ247123.1/132-52
AJ550900.1/56-3
M17439.1/1-48
AJ241823.1/282-335
M33000.1/55-110
M33000.1/55-110
M33000.1/55-110
AJ550900.1/56-3
Y12833.1/339-285
AJ550910.1/282-336
J02439.1/42-95
M17439.1/1-48
AJ247121.1/133-53
J02439.1/42-95
AJ005322.1/56-3
D00685.1/1-46
AJ005321.1/281-333
M63666.1/246-192
M33001.1/56-111
M17439.1/1-48
J02386.1/42-95
AJ550911.1/282-335
AJ550910.1/282-336
M33000.1/55-110
AJ005321.1/281-333
J02439.1/42-95
AJ247123.1/132-52
M33001.1/56-111
AJ005299.1/282-335
J02386.1/42-95
AF170516.1/283-335
M17439.1/1-48
M63666.1/246-192
AJ241850.1/282-334
AJ550911.1/282-335
M33001.1/56-111
AJ550900.1/56-3

AJ295015.1/58-1	AJ550899.1/56-3	AJ247123.1/132-52	D00685.1/1-46
AJ536614.1/206-152	AJ005303.1/56-3	AJ550906.1/282-334	AJ241845.1/282-335
AJ536617.1/1-40	AJ005319.1/56-3	AJ550910.1/282-336	M33000.1/55-110
AJ550911.1/56-3	AJ005300.1/282-335	AJ550910.1/282-336	M33001.1/56-111
AF170503.1/55-3	AJ005299.1/282-335	AJ005322.1/281-334	J02439.1/42-95
AJ536619.1/206-152	AJ550901.1/282-334	AJ241828.1/56-3	M33000.1/55-110
AJ005310.1/56-3	AJ550909.1/56-3	AJ005319.1/56-3	M33001.1/56-111
AJ241833.1/282-334	AJ550908.1/281-334	AJ550909.1/56-3	M63666.1/246-192
AF170504.1/284-337	AJ241847.1/281-334	AJ550898.1/282-335	M33000.1/55-110
AJ295018.1/58-1	AJ536619.1/206-152	AJ550912.1/56-3	AJ005314.1/281-334
AJ241833.1/282-334	AJ241847.1/281-334	AJ005300.1/282-335	M33001.1/56-111
AJ241841.1/57-3	AJ005320.1/281-333	AJ550907.1/56-3	M33000.1/55-110
AJ550901.1/282-334	AJ550911.1/282-335	AJ005294.1/282-334	M33001.1/56-111
AJ295018.1/58-1	AJ536619.1/206-152	AJ550906.1/56-3	Y12833.1/339-285
AJ536620.1/1-40	AJ247122.1/132-52	AJ550898.1/282-335	AJ005299.1/282-335

Acknowledgements

I gratefully acknowledge to Prof. Shunsuke Uemura and Prof. Kiyoshi Asai for all kinds of support and discussions. I also thank Yasuo Tabei, Michiaki Hamada, Taishin Kin and the other members of the functional RNA project for encouragement and discussions.

List of Publications

- Journal Papers (in English)

  1. Robust prediction of consensus secondary structures using averaged base pairing probability matrices.
     Hisanori Kiryu, Taishin Kin, and Kiyoshi Asai.*Bioinformatics*, accepted.(2006)
  2. Murlet: a practical multiple alignment tool for structural RNA sequences
     Hisanori Kiryu, Yasuo Tabei, Taishin Kin, and Kiyoshi Asai. submitted to *Bioinformatics* (2006)

- Oral Presentations (in Japanese)

  1. 塩基配列比較による機能性RNA発見法の開発
     木立　尚孝、田部井　靖生、金大真、浅井　潔. CBRC2006
  2. RNAのマルチプルアライメントからの二次構造予測について
     木立　尚孝、金　大真、浅井　潔. 新しいRNA/RNPを見つける会2006
  3. 二次構造を考慮した、機能性RNAのマルチプルアライメントプログラムの開発
     木立　尚孝、田部井　靖生、金大真、浅井　潔. 新しいRNA/RNPを見つける会2005

- Other Papers and Presentations

  1. Oscillations of persistent edge currents in the parafermion quantum Hall states.

H. Kiryu. *Physical Review B* 65, 113407.(2001)

2. Transcription rate of RNA polymerase under rotary torque.
   H. Kiryu. *Physical Review E* 69, 041902 (2004)

3. Extracting relations between promoter sequences and their strengths from microarray data.
   Hisanori Kiryu, Taku Oshima , and Kiyoshi Asai. *Bioinformatics* 21(7) 1062-8 (2005)

4. Regularities in the E. coli promoters composition in connection with the DNA strands interaction and promoter activity.
   Berezhnoy Andrey Yu , Shckorbatov Yuriy G. and Hisanori Kiryu. *Journal of Zhejiang Univ. Science B* 7 (12) 969-973 (2006)

5. 相互作用系におけるDisorderを取り扱うためのSuper　replica法
   木立　尚孝. 日本物理学会1998

6. Extracting Relations between Promoter Sequences and Their Strengths from Microarray Data.
   木立　尚孝. 第26回JSAI　SIGMBI分子情報研究会2003

References

[1] H. Carillo and D. Lipman. The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, 48(5):1073–1082, 1988.

[2] P. Carninci et al. The transcriptional landscape of the mammalian genome. *Science*, 309(5740):1559–63, 2005.

[3] C.B. Do, M.S.P Mahabhashyam, M. Brudno, and S. Batzoglou. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.*, 15(2):330–40, 2005.

[4] C.B. Do, D.A. Woods, and S. Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–8, 2006.

[5] R.D. Dowell and S.R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5:71, 2004.

[6] R.D. Dowell and S.R. Eddy. Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*, 7:400, 2006.

[7] A. Dunham et al. The DNA sequence and analysis of human chromosome 13. *Nature*, 428(6982):522–8, 2004.

[8] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis.* Cambridge University Press, 1998.

[9] S.R. Eddy. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, 3:18, 2002.

[10] P.P. Gardner, A. Wilm, and S. Washietl. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Res.*, 33(8):2433–2439, 2005. Evaluation Studies.

[11] J. Gorodkin, L.J. Heyer, and G.D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, 25(18):3724–32, 1997.

[12] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S.R. Eddy. Rfam: an RNA family database. *Nucleic Acids Res.*, 31(1):439–41, 2003.

[13] M. Hamada, K. Tsuda, T. Kudo, T. Kin, and K. Asai. Mining frequent stem patterns from unaligned RNA sequences. *Bioinformatics*, 2006.

[14] J.H. Havgaard, R.B. Lyngso, and J. Gorodkin. The FOLDALIGN web server for pairwise structural RNA alignment and mutual motif search. *Nucleic Acids Res.*, 33(Web Server issue):W650–3, 2005.

[15] D.S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18:341–343, 1975.

[16] I.L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res.*, 31(13):3429–31, 2003.

[17] I.L. Hofacker, S.H.F Bernhart, and P.F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20(14):2222–7, 2004.

[18] I.L. Hofacker, M. Fekete, C. Flamm, M.A. Huynen, S. Rauscher, P.E. Stolorz, and P.F. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucleic Acids Res.*, 26(16):3825–3836, Aug 1998.

[19] I.L. Hofacker, M. Fekete, and P.F. Stadler. Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, 319(5):1059–66, 2002.

[20] I.L. Hofacker, B. Priwitzer, and P.F. Stadler. Prediction of locally stable RNA secondary structures for genome-wide surveys. *Bioinformatics*, 20(2):186–90, 2004.

[21] I.L. Hofacker and P.F. Stadler. Automatic detection of conserved base pairing patterns in RNA virus genomes. *Compu.t Chem.*, 23(3-4):401–414, Jun 1999.

[22] I Holmes. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, 6:73, 2005.

[23] R. Knight, A. Birmingham, and M. Yarus. BayesFold: rational 2 degrees folds that combine thermodynamic, covariation, and chemical data for aligned RNA sequences. *RNA*, 10(9):1323–1336, Sep 2004.

[24] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, 31(13):3423–8, 2003.

[25] R. Luck, S. Graf, and G. Steger. ConStruct: a tool for thermodynamic controlled prediction of conserved secondary structure. *Nucleic Acids Res.*, 27(21):4208–4217, Nov 1999.

[26] R. Luck, G. Steger, and D. Riesner. Thermodynamic prediction of conserved secondary structure: application to the RRE element of HIV, the tRNA-like element of CMV and the mRNA of prion protein. *J. Mol. Biol.*, 258(5):813–826, May 1996.

[27] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288(5):911–40, 1999.

[28] D.H. Mathews and D.H. Turner. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J Mol Biol*, 317(2):191–203, 2002.

[29] B.W. Matthews. Comparison of predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451, 1975.

[30] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–19, 1990.

[31] S. Miyazawa. A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Eng.*, 8(10):999–1009, 1995.

[32] E.W. Myers and W. Miller. Optimal alignments in linear space. *Comput. Appl. Biosci.*, 4(1):11–7, 1988.

[33] R. Nussnov, G. Pieczenik, J.R. Griggs, and D.J. Kleitman. Algorithms for loop matchings. *SIAM J. Appl. Math.*, 35(1):68–82, 1978.

[34] Y. Okazaki et al. Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs. *Nature*, 420(6915):563–73, 2002.

[35] J.S. Pedersen, G. Bejerano, A. Siepel, K. Rosenbloom, K. Lindblad-Toh, E.S. Lander, J. Kent, W. Miller, and D. Haussler. Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput. Biol.*, 2(4):e33, 2006.

[36] J. Reeder and R. Giegerich. Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, 21(17):3516–3523, Sep 2005. Evaluation Studies.

[37] E. Rivas and S.R. Eddy. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2:8, 2001.

[38] D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45(5):810–825, 1985.

[39] Y. Tabei, K. Tsuda, T. Kin, and K. Asai. SCARNA: fast and accurate structural alignment of RNA sequences by matching fixed-length stem fragments. *Bioinformatics*, 22(14):1723–9, 2006.

[40] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22(22):4673–80, 1994.

[41] A.V. Uzilov, J.M. Keegan, and D.H. Mathews. Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. *BMC Bioinformatics*, 7:173, 2006.

[42] S. Washietl and I.L. Hofacker. Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.*, 342(1):19–30, 2004.

[43] S Washietl, I.L. Hofacker, M. Lukasser, A. Huttenhofer, and P.F. Stadler. Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. *Nat. Biotechnol.*, 23(11):1383–90, 2005.

[44] S. Washietl, I.L. Hofacker, and P.F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. U. S. A.*, 102(7):2454–9, 2005.

[45] M.S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *J. Mol. Biol.*, 197(4):723–728, Oct 1987.

[46] M. Zuker. Computer prediction of RNA structure. *Methods Enzymol.*, 180:262–88, 1989.