Bioinformation by Biomedical Informatics Publishing Group *open access*
www.bioinformation.net **Hypothesis**

Improving the prediction of RNA secondary structure by detecting and assessing conserved stems

Xiaoyong Fang[1, *], Zhigang Luo[1], Bo Yuan[2], Jinhua Wang[1]

[1]School of Computer Science, National University of Defense Technology, 410073 Changsha, China; [2]Department of Biomedical Informatics, College of Medicine and Public Health, Ohio State University, 43210-1239 Columbus Ohio, USA;
Xiaoyong Fang* - E-mail: xyfang@nudt.edu.cn; * Corresponding author

received December 14, 2007; accepted December 28, 2007; published online December 30, 2007

Abstract:
The prediction of RNA secondary structure can be facilitated by incorporating with comparative analysis of homologous sequences. However, most of existing comparative methods are vulnerable to alignment errors and thus are of low accuracy in practical application. Here we improve the prediction of RNA secondary structure by detecting and assessing conserved stems shared by all sequences in the alignment. Our method can be summarized by: 1) we detect possible stems in single RNA sequence using the so-called position matrix with which some possibly paired positions can be uncovered; 2) we detect conserved stems across multiple RNA sequences by multiplying the position matrices; 3) we assess the conserved stems using the *Signal-to-Noise*; 4) we compute the optimized secondary structure by incorporating the so-called reliable conserved stems with predictions by RNAalifold program. We tested our method on data sets of RNA alignments with known secondary structures. The accuracy, measured as sensitivity and specificity, of our method is greater than predictions by RNAalifold.

Keywords: RNA; secondary structure; conserved stem; homologous sequence; *Signal-to-Noise*

Background:

In recent years, RNAs gained increasing interest since a huge variety of functions associated with them were found **[1]**. It is now understood that RNA serves many cellular roles beyond being just a passive carrier of genetic information **[2, 3]**. The function of an RNA molecule is principally determined by its (secondary) structure. Unfortunately, the current physical methods available for structure determination are time-consuming and expensive **[4]**. For this reason, computational prediction provides an attractive alternative to facilitate the discovery of RNA secondary structure.

Minimum Free Energy (MFE) methods **[5]** and comparative sequences methods have been used to predict RNA secondary structure. However, there are several independent reasons why the accuracy of MFE structure prediction is limited in practice **[6]**. Generally, the best accuracy can be achieved by comparative methods **[7],** in which a large number of sequences are aligned to reveal the common base pairing pattern. So far, a number of methods based on comparative analysis of homologous sequences have been implemented to predict RNA secondary structure **[8-12]**. However, these approaches depend on fixed alignments and thus are very vulnerable to alignment errors. Several methods for simultaneous sequence alignment and structure prediction have been proposed to better this problem **[13-17]**. But algorithms based on these methods are too computationally taxing to be practical.

In this paper, we present a stem-based method to improve the prediction of RNA secondary structure. The central idea of our method is to detect and assess conserved stems shared by all sequences in the inputted RNA alignment, and then to compute the optimized secondary structure by incorporating reliable conserved stems (defined in Section Methods) with predictions by RNAalifold **[9]**. Our method improves RNAalifold by two major means: 1) to add some real base pairs which are possibly missed by RNAalifold for improving the sensitivity; 2) to remove some artificial base pairs which are possibly mistaken by RNAalifold for improving the specificity. We tested our method on data sets of RNA alignments taken from the Rfam database **[18]**. Experimental results suggest that our method can predict RNA secondary structure with much better performance than RNAalifold.

Methodology:

We presented the concept of position matrix for the first time in **[19]**. In this paper, we applied it to predicting RNA secondary structure.

Detecting possible stems in single sequence using the position matrix

We use the so-called position matrix to uncover some possibly paired positions in the sequence. Given an RNA sequence of length N, *Seq*, we build one $N \times N$ position matrix (denoted by M_{Seq}) by following steps: (1) The reverse complement of *Seq*, *Seq*', is firstly computed from the original sequence according to following rules: (a) The complement of 'G' is not 'C' but the set of {C, U}. For simplicity, we denote φ = {C, U}. (b) The complement of 'U' is not 'A' but the set of {A, G}. For simplicity, we denote ψ = {A, G}. (c) When the character is a gap (denoted by '_'), the complement for it should be a gap too. Here, the gaps in the sequence are possibly introduced by the inputted sequence alignment. (2) We build one $N \times N$ matrix (this is not the position matrix) containing *Seq*' in the first row. The ith ($0 \leq i \leq N$-1) row contains the sequence generated from *Seq*' by shifting i position to the left (circular left shift). (3) The position matrix M_{Seq} is computed by comparing *Seq* with the matrix generated by (2) row by row. 0 or 1 or -1 is assigned to the i, j ($0 \leq j \leq N$-1) element of M_{Seq} by comparing the jth character of *Seq* with the i, j ($0 \leq j \leq N$-1) element of the matrix of (2). Here, '0' means the corresponding position is unpaired and ungapped, '1' means the position is paired and ungapped, and '-1' means the position is gapped. The following rules should be obeyed when two characters (b and b') are compared with each other: (a) If b equals b' and neither of them is '_', then 1 is assigned. (b) If b or b' is '_' or both of them are '_', then -1 is assigned. (c) If b does not equal to b' and neither of them is '_', then 0 is assigned. Here, b is a character from *Seq* and b' is an element from the matrix of (2). Specially, when b' is φ or ψ, the word "equal" means that b belongs to b'. As shown in Figure 1, One $N \times N$ matrix (the left) is firstly built from the original sequence. Then the position matrix, M_{Seq}, is computed by comparing *Seq* with the left matrix row by row.

We can detect all possible stems in an RNA sequence by scanning the position matrix row by row. The key is to find all zones of continuous "1" in the matrix. There is a one-to-one mapping between the stems in the sequence and the zones of continuous "1" in the position matrix. As shown in Figure 1, two stems in the sequence (*Seq*) are mapped to two zones of continuous "1" in the position matrix (M_{Seq}).

When scanning one row of the position matrix (e.g. the ith row of M_{Seq}, $0 \leq i \leq N$-1), we divide the row into two parts: the elements for $0 \leq j \leq N$-1-i and the elements for N-$i \leq j \leq N$-1. Here, j is the column-subscript of the element. As for the former, we scan the elements backward from the (N-1-i)th column. As for the latter, we scan the elements forward from the (N-i)th column. Figure 2 is an example for the scanning of the position matrix, where M_{Seq} is the position matrix shown in Figure 1. Finally, we point that the time complexity of the approach mentioned above is O (N^2).

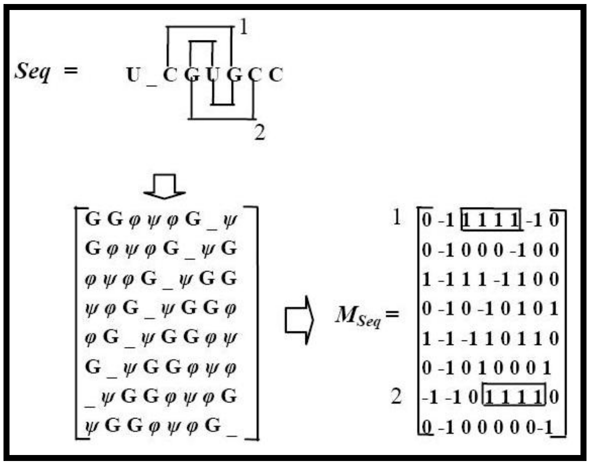


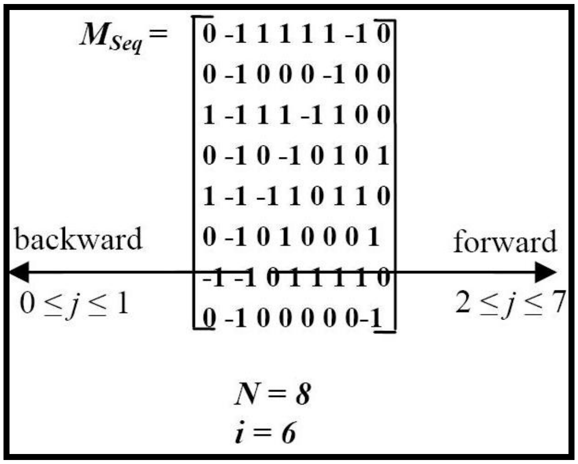
Figure 1: The construction of the position matrix for a gapped RNA sequence. The stem and its mapping zone are labelled by same number

Detecting conserved stems across multiple sequences by multiplying the position matrices

To detect conserved stems across multiple sequences, we first introduce the multiplying of position matrices. Suppose both M_1 and M_2 are $L \times L$ position matrices. Then the resulting matrix (denoted by M) for $M_1 \times M_2$ is computed by equation (1) (see supplementary material).

Specially, the multiplying of the elements must obey following rules: (1) $0 \times 0 = 0$; $0 \times 1 = 0$; $0 \times (-1) = 0$. (2) $1 \times 1 = 1$; $1 \times (-1) = 1$. (3) $(-1) \times (-1) = -1$.

The multiplying of n position matrices is computed by equation (2) (see supplementary material). Note that all original matrices for multiplying should have the same dimension. Obviously, the multiplying of the position matrices satisfies the commutative law and the associative law. We detect conserved stems shared by all sequences in an alignment by following steps: (1) We extract all sequences from the alignment and detect all possible stems in each sequence using the approach described in Figure 1 and Figure 2. (2) We select *n* different stems from *n* sequences (one stem for one sequence). Here, *n* is the number of sequences in the alignment. (3) We build the position matrix for each selected stem using the approach described in Figure 1 and multiply these *n* matrices according to the equation (2) (under supplementary material). (4) We detect conserved stems by finding zones of continuous '1' in the resulting matrix (*M*) generated by (3). There is still a one-to-one mapping between the conserved stems and the zones of continuous '1' in *M*. To find the zones of continuous '1', we scanning *M* using the approach described in Figure 2. (5) We repeat the steps (2) through (4) until all the stems detected by (1) are selected.



$$M_{Seq} = \begin{bmatrix} 0 & -1 & 1 & 1 & 1 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 1 \\ 1 & -1 & -1 & 1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Figure 2: As shown in the figure, the seventh row ($i = 6$) is divided into two parts: the elements for $0 \leq j \leq 1$ and the elements for $2 \leq j \leq 7$. As for the first part, this row is scanned backward from the second column (i.e. $j = 1$). As for the second part, this row is scanned forward from the third column (i.e. $j = 2$)

Specially, there are some problems about the step (3) when the selected stems have not the same length. In this case, for simplicity, we let the longest stem be unchanged and just add the gaps into the shorter stems at the beginning and the end. Then we build the position matrix for the new stems. As shown in Figure 3, we first select three different stems from the sequences in (a), and then detect the conserved stem (indicated by the rectangles in (b)) by scanning the matrix $M_{Stem1} \times M_{Stem2} \times M_{Stem3}$.

Actually, the equation (2) (in supplementary material) can be directly used to detect conserved stems in the RNA alignment. But in the approach mentioned above, we first extract all sequences from the RNA alignment and then to detect conserved stems shared by all sequences. The benefit for this is that some conserved stems possibly missed due to alignment errors also can be detected. Finally, we point that the time complexity of equation (2) (see supplementary material) is $O((n-1)L^2)$, where *L* is the dimension of the matrix. Suppose *m* stems are selected from each sequence on the average and the average length of the stem is still *L*. Then the time complexity for detecting conserved stems across *n* sequences of length *N* is approximately $O(m^n (n-1) L^2)$. In practical application, we keep *m* be a low constant to reduce the time cost. This can be easily done by removing some stems which are much shorter than some given length. Also, the condition $L<<N$ makes the time and space cost be low

Assessing conserved stems using the *Signal-to-Noise*

We use the *Signal-to-Noise* to assess the conserved stems. Here, the assessment means to determine whether the conserved stem belongs to a real RNA secondary structure or not. The major steps for assessing a conserved stem are: (1) We record the number of column pairs (see Figure 4) in the conserved stem as the Signal. (2) We generate a randomized alignment from the original alignment of the conserved stem. (3) We detect possibly conserved stems in the new alignment using the approach mentioned above. (4) We record the number of column pairs in the conserved stem newly detected by (3) as the Noise. The *Signal-to-Noise* is thus computed by Signal / Noise. We set Noise to be 1 if there are no conserved stems in the randomized alignment, and we set Noise to be the maximum if there are more than one conserved stems. Figure 4 is an example for computing the *Signal-to-Noise*.

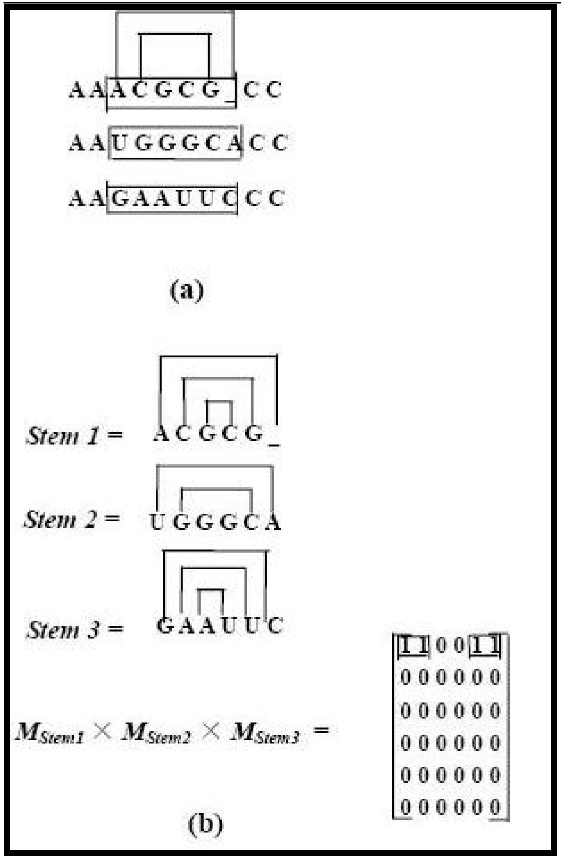


Figure 3: Finding conserved stems across multiple sequences by multiplying the position matrices: (a) is the sequence alignment, and (b) is for finding conserved stems shared by all sequences in the alignment

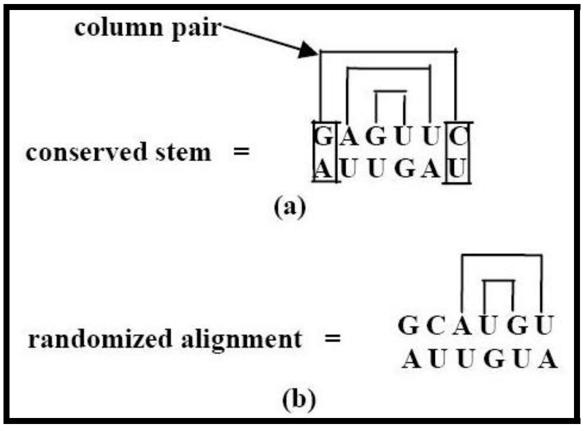


Figure 4: The computing of the *Signal-to-Noise*: (a) is the original alignment of the conserved stem, here the *Signal* is 3. (b) is the randomized alignment generated by permuting the columns of the original alignment, here the *Noise* is 2. As a result, the *Signal-to-Noise* is 3/2

The most difficulty in the approach mentioned above is how to generate a randomized alignment from the original alignment. To accomplish this purpose, we permute the columns of the original alignment until the common difference between the probability of the newest alignment and the probability of the last alignment is less than some given threshold (for example 0.001). To compute the probability of an RNA alignment, we introduce the so-called profile SCFG **[20]**. The profile SCFG can be defined by the five-tuple *Mol* = {*W*, *T*, *Al*, *E*}, where *Mol* is the profile SCFG model, *W* is the set of non-terminals, *T* is the set of transition distributions, Al is the set of terminals, and

E is the set of emission distributions. We define them as follows: (1) W = {start, bifurcation, single, pair, end}. (2) $T = [t\ (w,\ w')]$, where w and w' belong to W, and $t\ (w,\ w')$ is the transition probability from w to w'. (3) Al = {A, C, G, U, _}^{n}, where n is the number of sequences in the alignment and '_' symbolizes the gap. (4) $E = [e_w]$, where w belongs to W. If w is the non-terminal pair, then e_w should be $e\ (\beta,\ \beta')$. Here, both β and β' belong to Al, and they exactly form a column pair. If w is the non-terminal single, then e_w should be $e\ (\gamma)$. Here, γ is a single column which belongs to Al. Specially, the non-terminals, start, bifurcation and end do not produce any terminal.

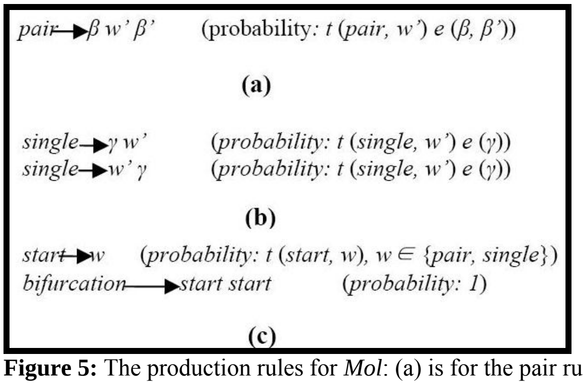
Specially, the profile SCFG presented here decomposes productions into two independent parts: non-terminal transitions and terminal emissions. The production rules can be categorized three classes: the pair rules, the single rules and the others. We describe them in Figure 5. The SCFG presented here has a great difference from other SCFGs, i.e., when a production rule is applied, a single column or a column pair not a single base or a base pair are generated. We use the inside-outside **[20]** method to estimate the parameters of the profile SCFG. And we change the original inside algorithm to compute the probability of an alignment. Actually, we use equation (3) and equation (4) (under supplementary material) to compute the probability of a derivation tree **[20]**.

Improving the prediction of secondary structure using reliable conserved stems

The central idea for improving structure prediction is first using RNAalifold to compute a basic structure and then using reliable conserved stems to revise it. Here, the reliable conserved stems refer to following two kinds of conserved stems: (1) The conserved stems with very high *Signal-to-Noise*. They are thought to belong to a real RNA secondary structure in our method. (2) The conserved stems with very low *Signal-to-Noise*. They are not thought to belong to any real RNA secondary structure in our method.

Specially, we should remove some incompatible conserved stems before determining the reliable conserved stems. For two conserved stems across multiple sequences, we say they are incompatible if at least one of following rules is satisfied: (1) They share at least one stem in the same sequence (see Figure 6a). (2) Suppose stem1 and stem2 are in the same sequence and they belong to different conserved stems. The position relationship of *stem1* and *stem2* is not consistent with the position relationship of the two conserved stems (see Figure 6b). For two incompatible conserved stems, we remain the one with greater *Signal-to-Noise* and remove the other. This is for selecting the first kind of reliable conserved stems. For selecting the second kind of reliable conserved stems, we can remain the one with lower *Signal-to-Noise* and remove the other. In this paper, the former is our selection.

In summary, we compute the optimized secondary structure for a given RNA alignment by following steps: (1) We compute a candidate secondary structure using RNAalifold, which is a popular and powerful program for predicting RNA secondary structure at present. (2) We extract all sequences from the RNA alignment and detect all possibly conserved stems shared by them. (3) We compute the *Signal-to-Noise* for each conserved stem. (4) We remove incompatible conserved stems and determine the reliable conserved stems according to the *Signal-to-Noise*. (5) We add the base pairs which are included by the first kind of reliable conserved stems but not included by the prediction by RNAalifold into the candidate secondary structure. (6) We remove the base pairs which are included by both the second kind of reliable conserved stems and the prediction by RNAalifold from the candidate secondary structure. (7) We compute the final secondary structure by merging the results of (5) and (6).



$pair \rightarrow \beta\ w'\ \beta'$ (probability: $t\ (pair,\ w')\ e\ (\beta,\ \beta')$)

(a)

$single \rightarrow \gamma\ w'$ ($probability$: $t\ (single,\ w')\ e\ (\gamma)$)

$single \rightarrow w'\ \gamma$ ($probability$: $t\ (single,\ w')\ e\ (\gamma)$)

(b)

$start \rightarrow w$ ($probability$: $t\ (start,\ w),\ w \in \{pair,\ single\}$)

$bifurcation \longrightarrow start\ start$ ($probability$: 1)

(c)

Figure 5: The production rules for *Mol*: (a) is for the pair rules. (b) is for the single rules. (c) is for the others

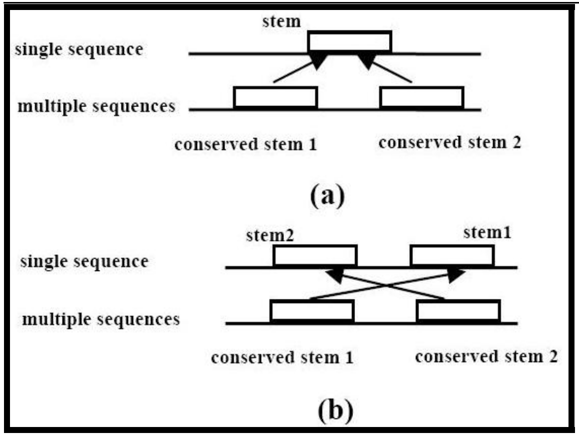


Figure 6: The condition for determining incompatible conserved stems; (a) is for the case that two conserved stems across multiple sequences share one same stem in one sequence. (b) is for the case that the position relationship of the two stems in one sequence is not consistent with the position relationship of the two conserved stems across multiple sequences

Results:

Test data sets

We implement the method using C++ programming language and test it on data sets constructed from Rfam 8.0 **[18]**. In more detail, we select 147 ncRNA families with sequence identity ranging from 40% to 99%. We respectively download the alignment of three sequences, four sequences and five sequences for each selected ncRNA family. In this way, we construct one data set of three-sequence alignments, one data set of four-sequence alignments and one data set of five-sequence alignments. Specially, the downloaded alignments are also used as the input for RNAalifold.

To test the performance of our method and compare it with RNAalifold, we also downloaded the consensus secondary structure annotated by Rfam 8.0 for each alignment. Specifically, Rfam 8.0 contains consensus secondary structures for each alignment either taken from a previously published study or predicted using some covariance-based methods. To make the test data set more reasonable, we remove some seed alignments with only predicted secondary structures. We measured the accuracy as the sensitivity and the specificity of predicted base pairs. Actually, we compute the sensitivity as the number of true positives divided by the sum of true positives + false negatives, and the specificity as the number of true positives divided by the sum of true positives + false positives.

Tests for multiple sequence alignments

We test our method on data sets constructed in Section Test data sets. The results are compared with RNAalifold, using the parameters suggested by the authors of **[9]**, and are reported in Table 1, Table 2 and Table 3 (see supplementary material for tables). The second and third columns in the tables are the results for our method. As shown in Table 1 (supplementary material), our method improves RNAalifold by 1.19% sensitivity and 4.21% specificity for three-sequence alignment tests. As shown in Table 2 (under supplementary material), our method improves RNAalifold by 2.07% sensitivity and 7.32% specificity for four-sequence alignment tests. As shown in Table 3 (in supplementary material), our method improves RNAalifold by 1.61% sensitivity and 4.65% specificity for five-sequence alignment tests. In general, our method has higher both sensitivity and specificity than RNAalifold.

One interesting thing about the results is that our method performs much better than RNAalifold for some ncRNA families. Actually, for the families of RF00012, RF00554, RF00046 and RF00094, our method can successfully detect more than 90% base pairs while RNAalifold can not correctly predict any base pair. We failed to get any valuable result about RNAalifold even though we tried to change some parameters of the program. On the other hand, our method sometimes exhibits little improvement to RNAalifold. In more detail, for the families of RF00521, RF00019, RF00557and RF00455, RNAalifold has 100% sensitivity and greater than 95% specificity and hence our method does not improve it at all.

Discussion:

Despite the limited amount of data, we have shown in the experiments that our method can predict RNA secondary structure with a better performance than RNAalifold do. Actually, our method adds some base pairs possibly missed by RNAalifold using the first kind of reliable conserved stems. This perhaps increases the true positives and thus leads a higher sensitivity. On the other hand, our method removes some base pairs possibly mistaken by RNAalifold using the second kind of reliable conserved stems. This perhaps decreases the false positives and thus leads to a higher specificity. Furthermore, the increased true positives contribute to a higher specificity, too.

In future work, our method can be improved by four means. One potential improvement could be computing the *Signal-*

to-Noise (defined in Methodology) in more effective ways. In this paper, we use the number of column pairs (defined in Methodology) in the conserved stem to accomplish this purpose. One alterative approach could be incorporating the free energy with the number of column pairs to compute it. Another way to improve our method might be determining the reliable conserved stems by more intelligent approaches. For example, the first kind of reliable conserved stems perhaps leads to increased false positives when it increases the true positives. Similarly, the second kind of reliable conserved stems perhaps leads to decreased true positives when it decreases the false positives. As a result, these reduce the accuracy of our method. Third, one perhaps concerns about the time complexity of the method, especially for the algorithm for generating randomized alignment. To better this problem, we can devise new parallel algorithms to speed up the method. But this needs further studying to remain the accuracy of the method. Finally, much longer sequences and more complicated structures should be benchmarked to further evaluate the performance of the method. Also, more existing comparative methods should be chosen to compare with the method.

Conclusion:

In this paper, we design, implement and evaluate a stem-based method for improving RNA secondary structure prediction. Our method detects conserved stems using the novel position matrix (defined in Methodology), assesses the conserved stems using the *Signal-to-Noise*, and improves RNAalifold using some reliable conserved stems. The fact that our method detects potential common stems shared by all sequences in the alignment can partly correct some prediction mistakes caused by alignment errors. Furthermore, the approach for detecting possible stems using the position matrix makes our method greatly differ from other methods. Finally, the approach for computing the optimized secondary structure using reliable conserved stems makes our method robust. As shown in the tests, our method can predict RNA secondary structure with much higher accuracy than RNAalifold program.

Acknowledgement:

This work has been supported by the National Natural Science Foundation of China under Grant No. 60673018.

References:

[01] J. Couzin, *Science,* 298: 2296 (2002) [PMID: 12493875]

[02] A. Huttenhofer, *et al.*, *TRENDS in Genetics*, 21: 289 (2005) [PMID: 15851066]

[03] S. R. Eddy, *Nat Rev Genet.,* 2: 919 (2001) [PMID: 11733745]

[04] B. Furtig, *et al.*, *Chembiochem.,* 4: 936 (2003) [PMID: 14523911]

[05] M. Zuker & P. Stiegler, *Nucleic Acids Research,* 9: 133 (1981) [PMID: 6163133]

[06] P. P. Gardner & R. Giegerich, *BMC Bioinformatics,* 5: 140 (2004) [PMID: 15458580]

[07] N. R. Pace, *The RNA World, 2nd ed.*, *NY: Cold Spring Harbor Laboratory Press*, 113 (1991)

[08] B. Knudsen & J. Hein, *Nucleic Acids Research,* 31: 3423 (2003) [PMID: 12824339]

[09] I. L. Hofacker, *et al.*, *Journal of Molecular Biology,* 319: 1059 (2002) [PMID: 12079347]

[10] J. Ruan, *et al.*, *Bioinformatics,* 20: 58 (2004) [PMID: 14693809]

[11] R. Knight, *et al.*, *RNA*, 10: 1323 (2004) [PMID: 15317972]

[12] Z. Weinberg & W. L. Ruzzo, *Bioinformatics,* 20: I334 (2004) [PMID: 15262817]

[13] D. Sankoff, *SIAM Journal on Applied Mathematics,* 45: 810 (1985)

[14] I. L. Hofacker, *et al.*, *Bioinformatics,* 20: 2222 (2004) [PMID: 15073017]

[15] D. Mathews & D. Turner, *Journal of Molecular Biology*, 317: 191 (2002) [PMID: 11902836]

[16] D. Mathews, *Bioinformatics,* 21: 2246 (2005) [PMID: 15731207]

[17] Y. Ji, *et al.*, *Bioinformatics,* 20: 1591 (2004) [PMID: 14962926]

[18] G. J. Sam, *et al.*, *Nucleic Acids Research,* 31: 439 (2003)

[19] X. Fang, *et al.*, *The 22nd Annual ACM Symposium on Applied Computing*, *Korea* (2007)

[20] R. Durbin, *et al.*, *Biological Sequence Analysis, Cambridge, Cambridge University press* (1998)

Edited by J. C. Tong, T. W. Tan & S. Ranganathan
Citation: Fang *et al.*, Bioinformation 2(5): 222-229 (2007)

Supplementary material

Equations

$M[i, j] = M_1[i, j] \times M_2[i, j]$ → (1)

Here, $M[i, j]$, $M_1[i, j]$ and $M_2[i, j]$ are respectively the i, j element of M, M_1 and M_2.

$$M = \prod_{i=1}^{n} M_i \quad \rightarrow \quad (2)$$

Here the resulting matrix is still denoted by M.

$$P(Tree|Mol) = \prod_{i=1}^{l_i} t_i(pair, w)e(\beta, \beta^{'})\prod_{j=1}^{l_2} t_j(\sin gle, w')e(\gamma)\prod_{k=1}^{l_3} t_k w, w' \in w \quad \rightarrow \quad (3)$$

Here, t_k is the probability of one rule from (c) of Figure 5, and *Tree* is the derivation tree which uses l_1 pair rules, l_2 single rules and l_3 others.

$$p(D|Mol) = \sum_{i=1}^{l_4} P(Tree_i|Mol) \quad \rightarrow \quad (4)$$

Here, D is the alignment which can be parsed by l_4 *derivation* trees in total given the *profile* SCFG, *Mol*.

Tables

Id (%)	*Se* (%)	*Sp* (%)	*Se*.RNAalif old (%)	*Sp*.RNAalif old (%)
<50	75.98	78.86	74.97	77.84
50-60	50.29	58.06	48.59	51.30
60-70	73.17	69.89	72.79	65.52
70-80	56.67	49.88	55.87	44.34
80-90	67.96	66.16	67.82	61.41
90-100	62.56	54.61	59.44	51.80
Total	64.44	62.91	63.25	58.70

Table 1: Sensitivity and specificity on data set of three-sequence alignments.

Id (%)	*Se* (%)	*Sp* (%)	*Se*.RNAalif old (%)	*Sp*.RNAalif old (%)
<50	75.16	89.96	74.60	86.84
50-60	48.65	50.43	43.48	44.48
60-70	74.88	59.21	74.56	56.64
70-80	65.01	50.11	61.90	41.51
80-90	73.18	67.11	71.51	64.10
90-100	50.28	51.03	48.70	41.22
Total	64.53	63.12	62.46	55.80

Table 2: Sensitivity and specificity on data set of four-sequence alignments.

Id (%)	*Se* (%)	*Sp* (%)	*Se*.RNAalif old (%)	*Sp*.RNAalif old (%)
<50	72.56	89.88	72.53	88.93
50-60	51.01	56.73	49.20	49.27
60-70	76.91	73.15	76.34	72.18
70-80	64.99	48.93	62.09	39.64
80-90	71.76	66.98	71.71	66.74
90-100	51.35	51.01	47.32	42.05
Total	64.76	64.45	63.15	59.80

Table 3: Sensitivity and specificity on data set of five-sequence alignments**.** (Id-percentage identity; *Se*-sensitivity; *Sp*-specificity)