BIOINFORMATICS Bioinformatics Advance Access published August 14, 2006

Mining frequent stem patterns from unaligned RNA sequences

Michiaki Hamada [a,b,c*], Koji Tsuda [a,d], Taku Kudo [e], Taishin Kin [a] and Kiyoshi Asai [a,f]

[a]Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-43 Aomi, Koto-ku, Tokyo, Japan, [b]Mizuho Information & Research Institute, Inc, 2-3, Kanda-Nishikicho,Chiyoda-ku,Tokyo 101-8443, Japan [c]Department of Computational Intelligence and System Science, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8503, Japan. [d]Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany. [e]Google Japan, Inc., 26-1, Sakuracho, Shibuya, Tokyo, 150-8512, Japan. [f]Graduate School of Frontier Sciences, University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa, Chiba, 277–8562, Japan.

Associate Editor: Golan Yona

ABSTRACT

Motivation: In detection of non-coding RNAs, it is often necessary to identify the secondary structure motifs from a set of putative RNA sequences. Most of the existing algorithms aim to provide the best motif or few good motifs, but biologists often need to inspect all the possible motifs thoroughly.

Results: Our method RNAmine employs a graph theoretic representation of RNA sequences, and detects all the possible motifs exhaustively using a graph mining algorithm. The motif detection problem boils down to finding frequently appearing patterns in a set of directed and labeled graphs. In the tasks of common secondary structure prediction and local motif detection from long sequences, our method performed favorably both in accuracy and in efficiency with the state-of-the-art methods such as CMFinder.

Availability: The software is available on request.

Contact: hamada-michiaki@aist.go.jp

Supplementary information: Visit the following URL for supplementary information, software availability and the information about the web server. http://www.ncrna.org/RNAMINE/.

1 INTRODUCTION

Recently, it is revealed that many RNAs, which are not translated into proteins, play essential roles at various biological stages. Those RNAs are called functional RNAs or non-coding RNAs (ncRNAs) and attracting remarkable attention. Computational and experimental screenings have predicted a number of non-coding RNAs (e.g., Deng *et al.*, 2006; Washietl *et al.*, 2005; Numata *et al.*, 2003), but only few of those RNAs are classified, because their functions are still unknown.

When a set of unaligned sequences of putative RNAs is provided without further information, we have to choose an appropriate analysis tool based on the *homogeneity* of the sequences. In this paper, we use the term *homogeneity* to both similarity of the sequences and that of the secondary structures. If the RNA sequences are highly homogeneous, they are evolutionarily related and share the unique common structure. In that case, the common structure can be predicted by RNAalifold (Hofacker *et al.*, 2002) or comRNA (Ji *et al.*, 2004), for example. Once the common structure has been determined, it can be used for a genome-wide scan by ,e.g., Infernal (Eddy and Durbin, 1994), RNAmotif (Macke *et al.*, 2001), or PHMMTS (Sakakibara, 2003).

The problem becomes more difficult when the homogeneity is low. In some cases, only the subset of the given sequences shares the common structure. In some cases, there are an unknown number of the clusters with different common structures. In order to analyze the sequences with low homogeneity, it is necessary to detect the secondary structure *motifs* shared by a significant fraction of the sequences, not by all. In order to find multiple motifs, it is possible to use a mixture of the probabilistic motif models (Blekas *et al.*, 2003) and train it using the EM algorithm (Dempster *et al.*, 1977), which is an algorithm for finding the maximum likelihood estimates of the parameters in the probabilistic models. However, that approach inevitably suffers from local minima problems, i.e., the solution is not guaranteed to converge to the global optimum. In this paper, we propose a new method based on a graph mining algorithm in order to detect the motifs shared by a subset of the given RNA sequences. Our method is also applicable to the set of sequences from multiple families, and able to find the multiple motifs. CMfinder (Yao *et al.*, 2005) does not assume that all the sequences have a common secondary structure, but are unable to find multiple motifs of the sequences from different families.

In this paper, an RNA sequence with its potential secondary structure is represented as a directed labeled graph, called a *stem graph*, each of whose node corresponds to a *stem candidate*. We employ *graph mining* algorithms, where highly probable motifs are *exhaustively* enumerated using the branch-and-bound algorithm over the well-designed data structures. Graph mining is a recently emerging subfield of data mining and a suite of algorithms are proposed recently (e.g., FSSM (Huan *et al.*, 2003), AGM (Inokuchi *et al.*, 2000, 2003), gSpan (Yan and Han, 2002)). Unlike the RNA graph proposed by Gan *et al.* (2003), it is not required that the secondary structure of each sequence is known in our algorithm. The nodes are made from the putative stems derived by thresholding McCaskill's base pairing probability matrix (Mathews, 2004). Therefore, the

[*] to whom correspondence should be addressed

stem graph can take into account all the possible secondary structures. A discrete label is assigned to each node such that the similar stem candidates share the common label. The labels are determined by a hierarchically clustering of all the stem candidates in the database. All the subgraphs appearing in at least m stem graphs, called *stem patterns*, are exhaustively enumerated. For the total number of graphs n, the fraction m/n is called *minimum support*. This parameter explicitly specifies the homogeneity of the sequence set. By setting minimum support to 0.9, for, example, we can enumerate all the stem patterns included in at least 90% of the sequences.

We developed a new graph mining algorithm by expanding gSpan (Yan and Han, 2002), because it turned out that conventional graph mining algorithms are too restrictive for our purpose. One problem is that patterns are identified based on the exact match of labels. In order to allow approximate label match, we introduce a *taxonomy* of the labels, which essentially describe the similarity of the labels. Furthermore, we exploited graph-theoretical properties of our RNA graphs to increase efficiency and reduce redundant solutions.

In the experiments, our algorithm will be applied to three different tasks. The first task is to predict the common secondary structure of every seed sequence in the specific Rfam (Griffiths-Jones *et al.*, 2005) families (Section 3.1). Since the sequence set is derived from a single family, the homogeneity is considered to be high. However, we will show that the accuracy of the prediction can be improved by exploiting multiple clusters in a family. The second task is to find two Rfam families in a mixed set of the sequences, where the minimum support is set to a small value (Section 3.2). Finally, a short motif will be found from a set of long RNA sequences (Section 3.3).

2 METHODS

Our main task is to find the frequent stem patterns from a database of sequences with unknown secondary structures. A core idea is to represent an RNA sequence as a new data structure called the *stem graph*. By the conversion of the sequences to the stem graphs, our task will be formulated as a mathematically well-defined problem. A motif is defined as a stem pattern in a graph-theoretic manner.

2.1 Stem Graphs and Stem Patterns

Let us begin with the stem graph of a sequence whose secondary structure is known. The known secondary structure of RNA can be represented by the set of stems in the structure. In a *stem graph*, a node corresponds to a stem and an edge between two nodes describes the relative position of the two corresponding stems (Figure 1, left). Each node is indexed by the three-tuple (S, d, p), where S is the stem sequence, d is the distance in nucleotide between $5'$ and $3'$ strands of the stem and p is the left-most position of the stem in the original RNA sequence. Each edge has a label P (Parallel), N (Nested) or K (Pseudoknotted) according to the relative position (Figure 2 left. See also Tabei *et al.* (2006)). An important feature is that when the secondary structure is known, the corresponding stem graph forms a *complete* graph (i.e., *clique*), since one of the three relations always applies to any pair of stems.

When the secondary structure is unknown, a stem graph is defined on the *stem candidates*, not on the confirmed stems as their nodes. The stem candidates are derived from the base pairing probability matrix calculated by McCaskill's algorithm (McCaskill, 1990) using the Vienna RNA package (Hofacker *et al.*, 1994) . The (i, j) value of this matrix represents the probability of the i-th nucleotide and the j-th nucleotide forming a base pair. Consecutive base pairs whose probability are more than p_{min} are identified as the part of the stem candidates, and the stem candidates shorter than l_{min} are discarded. The node index is expanded as (S, d, p, r), where r is the confidence of the stem, calculated as the average of base pairing probabilities. The stem graph made from a sequence is usually not complete, because there may be overlapping stems (Figure 2, right).

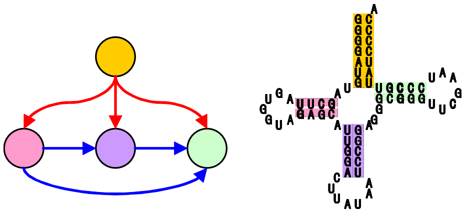

Fig. 1. An example of RNA sequence with known secondary structure (right: cited from Rfam database, http://www.sanger.ac.uk/Software/Rfam/) and its corresponding stem graph (left). The color of edge corresponds to a relation of stem indicated in left of Figure 2 and the color of the vertex in the left figure corresponds to the color of the stem in the right figure.

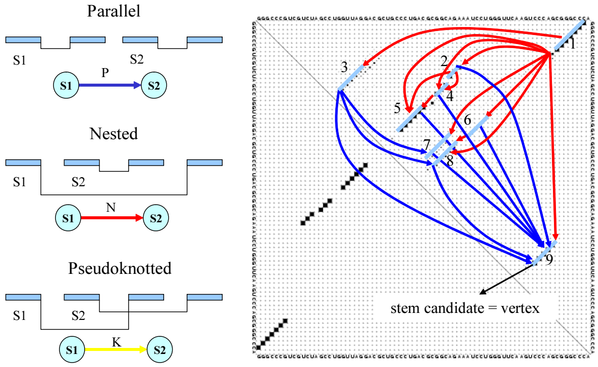

Fig. 2. Left: The three positional relations between two stem candidates. Right: An example of the stem graph superposed on the base pairing probability matrix of a tRNA sequence. A consistent RNA secondary structure must be a clique subgraph of this graph. For example $(1, 3, 7, 9)$ is consistent.

Since the node index has a complex form and not amenable to graph mining, it is translated into a set of discrete labels in the following way. The nodes of all the stem graphs are clustered and organized as a *label taxonomy*. First, a dendrogram is generated by the hierarchical clustering of the nodes (i.e., stem candidates) from *all* the stem graphs using the indices (S, d, p, r) (Figure 3, left), where the measure of dissimilarity will be presented in Section 2.2. Then, it is sliced to layers by the given dissimilarity thresholds. Then we generate a label for each cluster in each layer, and the resulting tree of labels is called the label taxonomy (Figure 3, right).

A *stem pattern* is represented as a directed labeled and clique graph where node labels are taken from arbitrary layers of the label taxonomy, and the edge label is either P,N or K. A stem pattern P *matches* to a directed graph G, if they have the same topology and the same edge labels, and every node label in P is an ancestor in the label taxonomy of the label of the corresponding node in G.

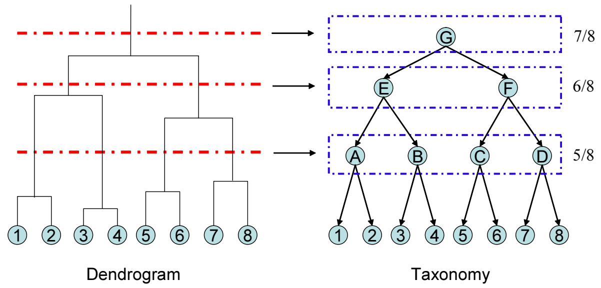

Fig. 3. Left: A dendrogram generated by a hierarchical clustering of the set of the stem candidates in *all* the sequences using dissimilarity described in 2.2 between stem candidates. Each leaf in left dendrogram indicates a stem candidate. Right: A label taxonomy constructed from the dendrogram in the left figure. The right most fractions of taxonomy are generalization cost of label in each layer. Taxonomy in this figure shows nodes 1 and 2 have the same label "A" in the 1st layer of taxonomy and 1, 2, 3, 4 have the same label "E" in the 2nd layer.

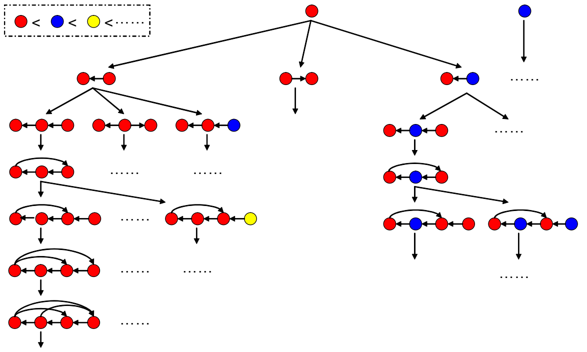

Fig. 4. Schematic figure of the tree-shaped search space of stem patterns. A child pattern is made by adding one edge to a previous pattern. The vertex labels have a predetermined order governing the extension process (shown as "$<$").

If a stem pattern finds matching subgraphs in some of the stem graphs, the corresponding RNA sequences share the partial common structure that are represented by the stem pattern.

2.2 Dissimilarity of Stems

We define the dissimilarity between two stems S_1 and S_2 as the weighted sum of the four components,

$$d(S_1, S_2) = \sum_{i=1,2,3,4} w_i d_i(S_1, S_2), \tag{1}$$

where w_i is the weight satisfying $\sum_{i=1,2,3,4} w_i = 1$. The first component accounts for the sequence similarity, $d_1(S_1, S_2) = \exp\{-\alpha SW(S_1, S_2)\}$, where $SW(S_1, S_2)$ is the score of the local alignment of the base pairs using RIBOSUM85-60 substitution matrix (Klein and Eddy, 2003). The confidence score of a stem $r(S)$ is calculated as the average of the base pairing probabilities. Based on those scores, the second component is derived as $d_2(S_1, S_2) = 1 - \frac{1}{2}(r(S_1) + r(S_2))$. Finally, the third and forth component are respectively computed as

$$d_3(S_1, S_2) = 1 - \frac{\beta + \min(d(S_1), d(S_2))}{\beta + \max(d(S_1), d(S_2))}$$

and

$$d_4(S_1, S_2) = 1 - \frac{\gamma + \min(p(S_1), p(S_2))}{\gamma + \max(p(S_1), p(S_2))}$$

using the loop distance (the distance between inner most base pairs) $d(S)$ and the start position (the position of leftmost base) of the stem $p(S)$. For the purpose of finding the local motifs from long sequences, w_4 should be set to zero in order to disregard the absolute positions. See (Ji *et al.*, 2004) and (Tabei *et al.*, 2006) for the alternative (dis)similarity measures.

2.3 Graph Mining with Label Taxonomy

In obtaining the stem patterns, we take a graph mining approach, where a set of constraints is determined first, and all the stem patterns satisfying the constraints are exhaustively enumerated. It is clearly different from other approaches that aim to obtain the best stem pattern. We set up the following three constraints. The first is to require that the stem pattern finds hits in at least m stem graphs, because it is meaningless to obtain a stem pattern with no match. Denote by $\mathrm{support}(P)$ the fraction of stem graphs that includes the pattern P. Then, the first constraint is written as $\mathrm{support}(P) \geq m/n$, where m/n is called the minimum support ($minsup$) later on. The second is that the stem pattern is constrained to be a clique to avoid overlapping stems. The third constraint is about the generality of the stem pattern. If many labels in the stem pattern are chosen from a higher layer of the taxonomy, the pattern is so general that the number of hits (i.e., subgraphs which matches to that pattern) is too large. To let the hits of the pattern form a meaningful RNA family, they have to be *similar* to each other. To encourage the use of the labels from low layers, the cost of a label is defined as an increasing function of the layer height. For a stem pattern P, the cost $\mathrm{cost}(P)$ is defined as the average cost of the labels of all its nodes. As the third constraint, we require $\mathrm{cost}(P)$ below a threshold, called the maximum cost ($maxcost$). In summary, our task is formulated as below.

FORMULATION 1. *Given a set of stem graphs, a label taxonomy, a minimum support $minsup$ and a maximum cost $maxcost$, completely enumerate every stem pattern P that satisfies the following conditions:*

1. $\mathrm{support}(P) \geq minsup$.
2. *P is a clique.*
3. $\mathrm{cost}(P) \leq maxcost$.

Comprehensive enumeration of all the subgraphs in a graph database is a well-studied subject in computer science. We have built our algorithm as an extension of a basic comprehensive algorithm called gspan (Yan and Han, 2002). Gspan is basically a branch-and-bound algorithm over a DFS code tree[1] (Figure 4). Each node of the tree is a DFS code which is a string representation of a graph. The tree is organized such that the child nodes represent the supergraphs of the parent node. The graphs in the database are enumerated by starting from the root node and expanding the tree by generating new child nodes. Yan and Han (2002) proposed an efficient way of generating

[1] DFS stands for depth first search.

the tree whose nodes enumerate the set of all the subgraphs of the original graph without redundancy.

In our problem, the stem patterns are defined as the subgraphs of the stem graphs satisfying the constraints. To impose the constraints, we focus on the following properties of the DFS code tree[2]. If the subgraph P_1 lies in the upstream of P_2 , then

- (a) $\mathrm{support}(P_1) \geq \mathrm{support}(P_2)$,
- (b) $\mathrm{cost}(P_1) \leq \mathrm{cost}(P_2)$ if P_1 and P_2 are cliques.

In our algorithm, the generation of the DFS code tree is restricted by exploiting those properties. For example, if a clique P whose support is below $minsup$ or cost is above $maxcost$ is found in the DFS code tree, we do not generate the downstream of P (i.e., tree pruning). Non-clique stem patterns are enumerated as well, but only the cliques are selected as the final solution. In comparison to the comprehensive enumeration, this restriction improves the efficiency and the memory consumption by orders of magnitude. The condition (a) is common in other graph mining methods, but the condition (b) is unique to our algorithm. Pseudocodes of our algorithms are shown in Algorithms 1 and 2. Algorithm 1 is the main part of our algorithm and Algorithm 2 is a recursively called subroutine. For technical details, see the supplementary paper.

Algorithm 1 RNAmine($RS, minsup, maxcost$)

```
Input: $RS$: set of RNA sequences, $minsup$: minimum support, $maxcost$: maximum cost
Output: $PS$: stem patterns which satisfy all the conditions in Formulation 1
$PS = \emptyset$
construct directed labeled graphs $GS$ and taxonomy $T$ from $RS$
$C_{initial} \leftarrow \{P : P$ is edge size $1, \mathrm{support}(P) \geq minsup$ and $\mathrm{cost}(P) \leq maxcost\}$
sort $C_{initial}$ in DFS lexicographic order
for all $s \in C_{initial}$ do
    Call GraphMining ($s, minsup, maxcost, GS, T, PS$)
end for
return $PS$
```

Algorithm 2 GraphMining ($s, minsup, maxcost, GS, T, PS$)

```
Input: $s$: current pattern, $GS$: directed labeled graph set, $T$: taxonomy of vertex label
if $\mathrm{support}(s) < minsup$ then return
if $\mathrm{cost}(s) > maxcost$ then return
if $s$ is a non-minimum DFS code then return
if $s$ is a clique then store pattern $s$ to $PS$
scan $GS$ once, find every edge $e$ that can be added to $s$ without violating the constraints; insert the found edges into $C$
sort $C$ in the DFS lexicographic order
for all $s \in C$ do
    Call GraphMining($s, minsup, maxcost, GS, T, PS$)
end for
```

2.4 Secondary structure prediction using stem patterns

Given a stem graph, any contained clique gives a consistent secondary structure without overlapping stems. The *minimum free energy* (MFE) of the corresponding secondary structure of such a clique can be computed by a software like RNAeval in Vienna RNA package(Hofacker *et al.*, 1994).

Using the MFE of the cliques, the secondary structure prediction is done in the following way. Assume that the stem patterns are already obtained from a set of RNA sequences. In predicting the secondary structure of a sequence in the set, we first identify the set of stem patterns that have matching subgraphs. A matching subgraph corresponds to a predicted secondary structure. Notice that one stem pattern can be matched in a several different ways, creating slightly different secondary structures (Figure 7). Repeating this procedure through all the stem patterns, the structures of all the sequences are determined. This process gives a number of secondary structures to one sequence, so they are ranked by their minimum free energy (Figure 7).

3 RESULTS

For benchmarking, we used the Rfam database (version 7.0, March 2005) containing 503 RNA families (Griffiths-Jones *et al.*, 2005), whose common secondary structures are available for the seed sequences. We selected eight families whose number of hairpins is more than three. All the families except tRNA are used by Yao *et al.* (2005) as well. For the families that contain more than 50 seed sequences, we randomly selected 50 sequences. We did not include the sequences with a nucleotide character other than A, C, U, G and T. The dataset is summarized in Table 1. All experiments are performed using a machine with a 2.4GHz AMD Opteron™ processor and 20GB memory.

3.1 Secondary Structure Prediction

Secondary structure prediction of an individual sequence can be done by free energy minimization using, e.g., RNAfold (Hofacker *et al.*, 1994). However, when the sequences share a common secondary structure, it is often better to find the common structure and parse each sequence using the common structure. This process can be implemented using the EM algorithm over the covariance model, CMfinder (Yao *et al.*, 2005) and a graph-theoretical method, comRNA (Ji *et al.*, 2004). When a multiple alignment of the sequences is given a priori, the secondary structure can be predicted by, e.g., RNAalifold (Hofacker *et al.*, 2002). Another approach to predict the secondary structure is to derive a number of *suboptimal* secondary structures for a sequence e.g.,Wuchty *et al.* (1999).

In this experiment, RNAmine is compared with CMfinder (Yao *et al.*, 2005), RNAfold (Hofacker *et al.*, 1994), RNAsubopt (Wuchty *et al.*, 1999), RNAalifold (Hofacker *et al.*, 2002), and comRNA (Ji *et al.*, 2004). CMfinder and comRNA exploit the common secondary structure, while RNAfold and RNAsubopt predicts the structure individually. Like RNAmine, RNAsubopt derives a number of multiple possible structures for a sequence. RNAalifold assumes the multiple alignment, which is made here by clustalW (Thompson *et al.*, 1994). Those tools are used mostly with the default parameters (See the supplementary paper for details). For RNAmine we set $minsup$ and $maxcost$ to be 0.7 and 0.6, and w_1, w_2, w_3 and w_4 to be 0.6, 0.15, 0.10, 0.15, respectively.

[2] Our DFS code tree is slightly different from original one. See supplementary paper for details.

Table 1. Summary of the test data and the results

Family	RFAM_ID	#seqs	length	%id	RNAmine			CMfinder	comRNA	RNAalifold	RNAfold	RNAsubopt	
					1	5	10	MCC	MCC	MCC	MCC	MCC	str/seq
Cobalamin	RF00174	50	203.2	43	0.41	0.52	0.53	0.54	0.00	0.47	0.34	0.44	119.6
Lysine	RF00168	50	181.6	46	0.80	0.85	0.86	0.79	0.21	0.35	0.64	0.74	112.3
Purine	RF00167	37	99.6	53	0.83	0.90	0.91	0.89	0.00	0.52	0.73	0.81	8.3
RFN	RF00050	48	137.2	64	0.62	0.71	0.74	0.41	0.00	0.57	0.44	0.52	29.4
S_box	RF00162	50	110.4	61	0.77	0.82	0.84	0.78	0.29	0.48	0.64	0.76	35.1
Tymo_tRNA-like	RF00233	27	82.6	66	0.76	0.88	0.88	0.93	0.55	0.51	0.60	0.72	10.3
glmS	RF00234	14	177.6	55	0.80	0.86	0.90	0.88	0.47	0.35	0.58	0.66	30.7
tRNA	RF00005	50	73.4	40	0.75	0.84	0.84	0.78	0.00	0.37	0.60	0.73	8.5
				average	0.72	0.80	0.81	0.75	0.19	0.45	0.57	0.67	44.3

RFAM_ID: ID number in Rfam database (http://www.sanger.ac.uk/Software/Rfam/). #seq: the number of sequences in each family. length: average length of sequence in each family. %id: average sequence identity calculaed by alistat program. MCC: average MCC among sequences. Best MCC among top 1, 5 and 10 structures are shown in result of RNAmine. For comRNA and RNAsubopt, the best MCC among predicted common secondary structures is shown (if comRNA produced no motif, MCC is 0 in this table). str/seq (for RNAsubopt): the average number of predicted suboptimal secondary structures per sequence. The definition of MCC is found in the supplementary paper.

We used the MCC (Mathews Correlation Coefficient), defined in section 3 in the supplementary paper, as the performance measure. The average MCCs are summarized in Table 1 and the running times are shown in Table 2. See sensitivity and PPV for each family in the supplementary paper (Table S1 and S2). The number of predicted structures per sequence is also shown for RNAsubopt. RNAmine performed better than RNAfold, RNAsubopt and comRNA in most cases. The accuracies of RNAfold are not better than those of the other methods, showing the difficulty of predictions from individual sequences. The results of RNAalifold in Table 1 have limited accuracies because the alignments of clustalW were used. When the reference alignments that had been annotated in Rfam database were used, the results were much better (See Table S3 in supplementary paper). RNAsubopt performed relatively well, but it is due to the large number of predictions (about 120 in maximum and 44 in average). In comparison to CMfinder, RNAmine achieved comparative accuracy overall, and for several families such as Lysine and RFN, RNAmine performed better. The homogeneity of this dataset was relatively high, because the sequence set is derived from one family. This result shows that RNAmine can compete well with the state-of-the-art methods even in those clean datasets. In addition, RNAmine can deal with non-homogeneous data as shown in the next section. See Figure 7 for example of the actual predicted secondary structures of a sequence in Tymo_tRNA-like family.

It is remarkable that RNAmine was more than twenty times faster than CMfinder, though the worst-case time complexity of graph mining is theoretically NP-hard (Inokuchi, 2004). Graph mining is fast when the size of search tree is kept small as in our implementation.

Figure 5 shows the best accuracies among top-ranked structures. The accuracy saturates around rank 10, implying that one needs to inspect only top ten structures. Figure 6 illustrates the change of accuracy and computation time against the minimum support parameter. The computational time decreases monotonically as the minimum support increases, because the search tree can be pruned earlier if the minimum support is high. It is interesting to see that the best accuracy is achieved at 0.7, which is much better than the accuracy at 1. Setting the minimum support to 1, a stem pattern matching all the sequences is obtained. However, when the minimum support is below one, multiple stem patterns are obtained, each of which matches to a subset of sequences. This result shows the structure prediction can be enhanced by exploiting hidden clusters in the family.

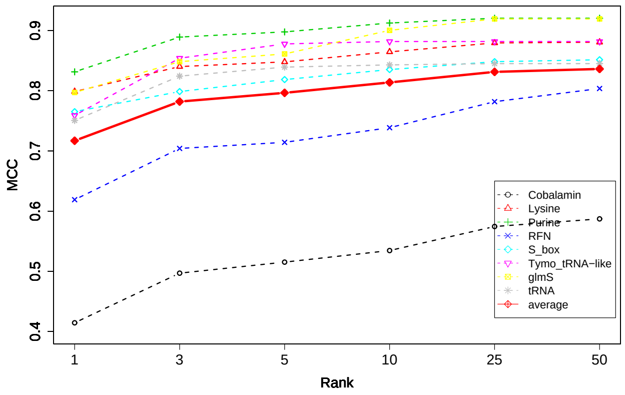

Fig. 5. The best MCC of RNAmine among top-ranked secondary structures. Each dashed-line indicates the MCC for individual family and the solid line shows the average accuracy.

3.2 Dataset with Multiple Families

In this experiment, our method is applied to the input sequences including *multiple* RNA families. Six datasets are generated by combining two families in Table 1 into one. We compared proposed method with only RNAfold and RNAsubopt, because the other tools assume that input sequences are related sequences or cannot handle multiple families. For RNAmine, $minsup$ is set to be 0.3 and the other parameter settings are the same as the previous experiments in Table 1. Table 3 shows our results. In comparison to RNAfold and RNAsubopt, RNAmine has achieved better accuracies uniformly in all the datasets. Moreover, RNAmine indeed detected the two families as the separate stem patterns in most cases (see the supplementary paper).

```
>Y16104.1/6590-6672
UAAUUGAGGACAGUUCCUCUCCCUCUAGCACACAGAGGUCAAACUGGGUGCAACUCCCCCCCCUUCCGUGGGUAACGGAAACC
.....aaaaa....aaaaa..bbbbb.......bbbbb.......ccc.......ccc......ddddd.....ddddd....
.....(((((....)))))..(((((.......))))).......(((.......))).....((((((.....))))))...  -21.60
.....(((((....)))))..(((((.......))))).......(((........)))....((((((.....))))))...  -21.40
.....(((((....)))))..(((((.......))))).......(((.........)))...((((((.....))))))...  -20.60
.....(((((....)))))..(((((.......))))).......(((..........)))..((((((.....))))))...  -20.50
.....(((((....)))))..(((((.......))))).......(((...........))).((((((.....))))))...  -20.30
```

Fig. 7. An example of predicted secondary structures for a sequence in Tymo_tRNA-like family. Top-ranked secondary structures are shown. The free energy is shown at the end of each structure. Lower case letters indicate the correct stems given by Rfam annotations.

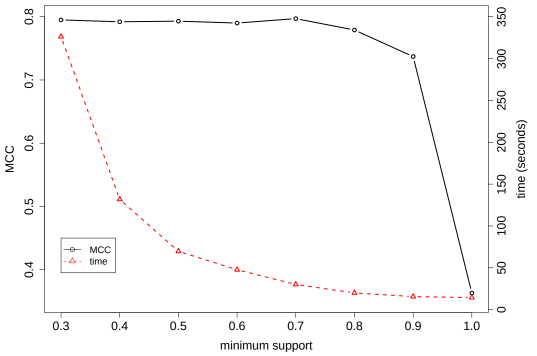

Fig. 6. The average MCC (best of top 5) and average calculation time among 8 families in Table 1. Solid and dashed lines indicate MCC and time, respectively. We set $maxcost$ to be 0.6.

Table 2. Running time (seconds) for each family in Table 1

Family	RNAmine	CMfinder	comRNA	RNAsubopt
Cobalamin	46.7	1684.9	1072.1	20.3
Lysine	115.7	1397.5	944.5	19.3
Purine	4.0	198.9	1216.1	0.6
RFN	35.5	630.8	648.7	2.8
S_box	8.0	352.6	627.2	2.8
Tymo_tRNA-like	1.8	106.0	816.7	0.3
glmS	26.9	355.3	732.5	1.8
tRNA	2.5	230.5	1220.4	0.6
average	30.1	619.6	909.8	6.0

The results of RNAalifold and RNAfold are omitted. The running time of both tools are within a few seconds for all the families.

3.3 Local motif detection from long RNA sequences

Recently many long (more than 1000 bases) non-coding RNAs, called mRNA-like non-coding RNAs, are detected by genome-wide analysis of cDNAs, e.g., (Numata *et al.*, 2003). In this experiment, our algorithm is applied to detect motifs from a family called BIC whose secondary structure motif has already been reported (Tam, 2001). BIC is a microRNA host gene and T-cell activation early gene (van den Berg *et al.*, 2003). Our dataset is prepared by using Tam's paper and Regulatory non-coding RNAs database (http://biobases.ibch.poznan.pl/ncRNA/). Our BIC dataset has three sequences (Human, Mouse, Chicken). The average sequence length is 1715 and the average sequence similarity is 53%. In this experiment, we set $minsup$ to be 1.0 because we would like to find the common motifs in all the sequences. Also we set the maximum motif size to 100 for detecting the *locally* conserved motifs.[3] We selected the stem patterns of maximum size (4 in this case) and, among them, the best motif is identified as the one with the minimum cost (Figure 8). Magenta, blue and green stems in Figure 8 correspond to the reported stems I, II and V, respectively (see Figure 9). This result highlights RNAmine's ability of detecting local motifs from a small dataset.

Table 3. Results for multiple family dataset

	RNAmine	RNAfold	RNAsubopt	
Family	MCC	MCC	MCC	#secs/seq
Cobalamin + Lysine	0.62	0.49	0.59	116
Lysine + RFN	0.79	0.54	0.63	70.9
Purine + Tymo_tRNA-like	0.91	0.67	0.76	9.3
S_box + Purine	0.86	0.69	0.78	21.7
tRNA + S_box	0.79	0.62	0.75	21.8
tRNA + Tymo_tRNA-like	0.82	0.60	0.73	9.4
average	0.80	0.60	0.71	41.5

Each dataset is created by combining two families in Table 1 into one dataset. MCC: average MCC. For RNAsubopt, the best MCC among predicted suboptimal secondary structures is shown. For RNAmine, the best MCC among top 5 structures is shown. #secs/seq (for RNAsubopt): the average number of predicted secondary structures per sequence.

4 DISCUSSION AND CONCLUSION

We have developed a novel algorithm for mining stem patterns from RNA sequences by extending graph mining techniques. One of the remarkable points of our approach is that multiple motifs can be found in a set of sequences from multiple RNA families. The homogeneity of given sequences can be explicitly specified by the parameter $minsup$. The effectiveness of our algorithm was confirmed by comparing with the other secondary structure prediction tools and detecting the local motifs from long RNA sequences. Although the search space has been reduced by adopting a minimum support and a maximum cost of the stem pattern, the worst computational complexity is not polynomial order. Considerably longer computational time is required for longer sequences or larger data sets (see Figure S4 and S5 in supplementary paper). For

[3] That is realized by not making edges between two stem candidates more than 100 bases away from each other.

```
>Hs_bic
225 GUAGGCUGUAUGCUGUUAAUGCUAAUCGUGAUAGGGGUUUUUGCCUCCAACUGACUCCUACAUAUUAGCAUUAACAGUGUAUGAUGCCUGU 315
225 ((((((((((((((((((((((((((.....((((((((.(((....)))..))))))))...))))))))))))))))))))..)))))) 315
>Mm_bic
148 AGGCUGUAUGCUGUUAAUGCUAAUUGUGAUAGGGGUUUUGGCCUCUGACUGACUCCUACCUGUUAGCAUUAACAGGACACAAGGCCU 234
148 (((((.....((((((((((((((.....((((((((((((...))))..))))))))...)))))))))))))).......))))) 234
>Gg_bic
343 AGGCUGUAUGUUGUUAAUGCUAAUCGUGAUAGGGGUUUUUACCUCUGAAUGACUCCUACAUGUUAGCAUUAACACUGUACCAUGCCU 429
343 ((((.......(((((((((((((.....((((((((((((....)))).))))))))...))))))))))))).........)))) 429
```

Fig. 8. A motif of BIC found by RNAmine. Red, green and magenta stems correspond to the reported stems I, II and V in Figure 9, respectively. Left number of sequence indicates the start point in mother sequence and right does the end point.

long sequences, local search version of RNAmine can reduce its computational time by ignoring the remote base pairs that have distances longer than a threshold. The local search can be implemented in RNAmine by using local base pairing probabilities in RNAplfold Bernhart *et al.* (2006). Our parameters (e.g., p_{min}) are optimized manually using small data sets. Automatic optimization of the parameters is one of our future works. Our recommendation of the parameter p_{min} is from 0.001 to 0.01 (0.05 is used in our experiments) and the parameter l_{min} is from 3 to 5 (3 is used in our experiments).

Due to our graph representation, our method can deal with *pseudoknotted* structures, unlike CMfinder, RNAalifold and RNAsubopt.[4]. The efficiency of our algorithm would be further improved by the following ideas: (a) Constraining the smallest cost of vertex labels for the *anchoring* effect (Touzet and Perriquet, 2004). (b) To Avoid enumerating *over-generalized patterns* (Inokuchi, 2004). (c) Enumerating only *closed patterns* (Yan and Han, 2003). To apply our algorithm to large scale motifs and cluster detection problems, more elaborations might be needed on the scoring method of stem patterns, though a simple scoring scheme performed well in a small dataset in Section 3.3.

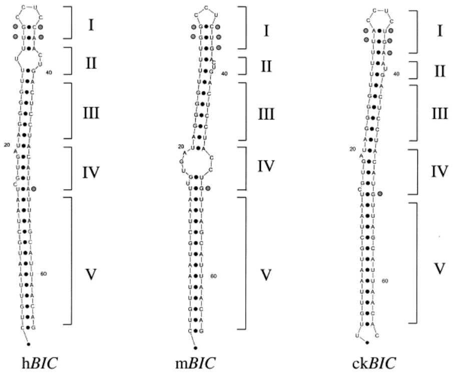

Fig. 9. The known secondary structure motifs of BIC reported in Tam (2001). This figure is cited from (Tam, 2001)

5 ACKNOWLEDGMENTS

This work is partially supported by "Functional RNA Project" of New Energy Technology Development Organization. The authors thank Akihiro Inokuchi, Hisanori Kiryu, Tsuyoshi Kato, Toshiyuki Sato, Suenori Chiku and the other members of bioinformatics group for RNAs at National Institute of Advanced Industrial Science and Technology (AIST) and at Japan Biological Information Consortium (JBIC) for fruitful discussions. The web server of RNAmine has been developed by Axiohelix, Japan. Thanks are also due to two anonymous referees for useful suggestions.

REFERENCES

Bernhart, S. H., Hofacker, I. L. and Stadler, P. F. (2006) Local RNA base pairing probabilities in large sequences. *Bioinformatics*, **22**, 614–615.

Blekas, K., Fotiadis, D. and Likas, A. (2003) Greedy mixture learning for multiple motif discovery in biological sequences. *Bioinformatics*, **19**, 607–617.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). **39**, 1–38.

Deng, W., Zhu, X., Skogerbo, G., Zhao, Y., Fu, Z., Wang, Y., He, H., Cai, L., Sun, H., Liu, C., Li, B., Bai, B., Wang, J., Jia, D., Sun, S., He, H., Cui, Y., Wang, Y., Bu, D. and Chen, R. (2006) Organization of the Caenorhabditis elegans small non-coding transcriptome: genomic features, biogenesis, and expression. *Genome Res*, **16**, 20–29.

Dirks, R. M. and Pierce, N. A. (2003) A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry*, **24**, 1664–1677.

Eddy, S. R. and Durbin, R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res*, **22**, 2079–2088.

Gan, H., Pasquali, S. and Schlick, T. (2003) Exploring the repertoire of RNA secondary motifs using graph theory: Implications for RNA design. *Nucleic Acids Res.*, **31**, 2926–2943.

Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R. and Bateman, A. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*, **33**, 121–124.

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, S., Tacker, M. and Schuster, P. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.

Hofacker, I. L., Fekete, M. and Stadler, P. F. (2002) Secondary structure prediction for aligned RNA sequences. *J Mol Biol*, **319**, 1059–1066.

[4] In construction of stem graphs, we used the McCaskill's algorithm (McCaskill, 1990) that basically does not consider the pseudoknots in the calculation of base pairing probabilities. But this part is not essential and can be replaced by another algorithm (Dirks and Pierce, 2003)

Huan, J., Wang, W. and Prins, J. (2003) Efficient mining of frequent subgraphs in the presence of isomorphism. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, p. 549. IEEE Computer Society, Washington, DC, USA.

Inokuchi, A. (2004) Mining generalized substructures from a set of labeled graphs. In *ICDM*, pp. 415–418. IEEE Computer Society.

Inokuchi, A., Washio, T. and Motoda, H. (2000) An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 13–23. Springer-Verlag, London, UK.

Inokuchi, A., Washio, T. and Motoda, H. (2003) Complete mining of frequent patterns from graphs: Mining graph data. *Mach. Learn.*, **50**, 321–354.

Ji, Y., Xu, X. and Stormo, G. D. (2004) A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, **20**, 1591–1602.

Klein, R. J. and Eddy, S. R. (2003) RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics*, **4**, 44.

Macke, T. J., Ecker, D. J., Gutell, R. R., Gautheret, D., Case, D. A. and Sampath, R. (2001) RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res*, **29**, 4724–4735.

Mathews, D. H. (2004) Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*, **10**, 1178–1190.

McCaskill, J. S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Numata, K., Kanai, A., Saito, R., Kondo, S., Adachi, J., Wilming, L. G., Hume, D. A., Hayashizaki, Y. and Tomita, M. (2003) Identification of putative noncoding RNAs among the RIKEN mouse full-length cDNA collection. *Genome Res*, **13**, 1301–1306.

Sakakibara, Y. (2003) Pair hidden Markov models on tree structures. *Bioinformatics*, **19 Suppl 1**, 232–240.

Tabei, Y., Tsuda, K., Kin, T. and Asai, K. (2006) SCARNA:Fast and Accurate Structural Alignment of RNA Sequences by Matching Fixed-length Stem Fragments. *Bioinformatics Advance Access*. doi:10.1093/bioinformatics/btl177.

Tam, W. (2001) Identification and characterization of human BIC, a gene on chromosome 21 that encodes a noncoding RNA. *Gene*, **274**, 157–167.

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994) 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**, 4673–4680.

Touzet, H. and Perriquet, O. (2004) CARNAC: folding families of related RNAs. *Nucleic Acids Res*, **32**, 142–145.

van den Berg, A., Kroesen, B.-J., Kooistra, K., de Jong, D., Briggs, J., Blokzijl, T., Jacobs, S., Kluiver, J., Diepstra, A., Maggio, E. and Poppema, S. (2003) High expression of B-cell receptor inducible gene BIC in all subtypes of Hodgkin lymphoma. *Genes Chromosomes Cancer*, **37**, 20–28.

Washietl, S., Hofacker, I. L., Lukasser, M., Huttenhofer, A. and Stadler, P. F. (2005) Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. *Nat Biotechnol*, **23**, 1383–1390.

Wuchty, S., Fontana, W., Hofacker, I. L. and Schuster, P. (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.

Yan, X. and Han, J. (2002) gspan: Graph-based substructure pattern mining. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, p. 721. IEEE Computer Society, Washington, DC, USA.

Yan, X. and Han, J. (2003) Closegraph: mining closed frequent graph patterns. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 286–295. ACM Press, New York, NY, USA.

Yao, Z., Weinberg, Z. and WL, W. R. (2005) CMfinder–a covariance model based RNA motif finding algorithm. *Bioinformatics*, **22**, 445–452.