IPSJ Transactions on Bioinformatics Vol. 2 36–46 (Mar. 2009)

Original Paper

Large Scale Similarity Search for Locally Stable Secondary Structures among RNA Sequences

Michiaki Hamada,[†1,†2,†3] Toutai Mituyama[†2] and Kiyoshi Asai[†2,†4]

Recently, a large number of candidates of non-coding RNAs (ncRNAs) has been predicted by experimental or computational approaches. Moreover, in genomic sequences, there are still many interesting regions whose functions are unknown (e.g., indel conserved regions, human accelerated regions, ultraconserved elements and transposon free regions) and some of those regions may be ncRNAs. On the other hand, it is known that many ncRNAs have characteristic secondary structures which are strongly related to their functions. Therefore, detecting clusters which have mutually similar secondary structures is important for revealing new ncRNA families. In this paper, we describe a novel method, called RNAclique, which is able to search for clusters containing mutually similar and locally stable secondary structures among a large number of unaligned RNA sequences. Our problem is formulated as a constraint quasi-clique search problem, and we use an approximate combinatorial optimization method, called GRASP, for solving the problem. Several computational experiments show that our method is useful and scalable for detecting ncRNA families from large sequences. We also present two examples of large scale sequence analysis using RNAclique.

1. Introduction

Recent research has revealed a number of RNAs which are not translated into protein but nevertheless play an important role in cells. These RNAs are called non-coding RNAs (ncRNAs for short) or functional RNAs and have attracted much attention [4)–6),10),15),21),22),26),30),34)]. Computational or experimental approaches predict a huge number of ncRNA candidates in the human genome. Using computational screening, Washietl, et al. [31)] predicted about 35,000 ncRNA candidates which are structurally conserved and thermodynamically stable RNA secondary structures in a multiple sequence alignment, and Pedersen, et al. [26)] found about 40,000 candidates using phylogenetic stochastic context-free grammars (phylo-SCFGs), which are combined probabilistic models of RNA secondary structure and primary sequence evolution. Experimentally, Nakaya, et al. [24)] have reported 55,139 totally intronic noncoding (TIN) RNAs transcribed from the introns and 12,592 partially intronic noncoding (PIN) Expressed Tag Sequencing (EST) contigs. More recently Kapranov, et al. [16)] used a tiling array and reported more than 1 million transcriptional fragments. Moreover, there are still many interesting regions in (human) genomes, such as ultraconserved elements [2)], human accelerated regions [28)], indel-conserved regions [20)], and transposon free regions [29)]. The functions of most of these sequences or regions are not yet known, and those regions may include novel ncRNAs or ncRNA families. In this paper, we focus on detecting ncRNA families from those regions.

It is well known that the function of most ncRNAs is strongly related not only to their sequences, but also to their *secondary structures*, which are sets of base pairs that occur in its three-dimensional structure. (We consider only canonical (A-U and G-C) and wobble (G-U) base pairs in this paper.) There exists a number of functional families which have similar secondary structures, such as tRNA, snoRNA, and 7SK RNA. In the Rfam database [8)] various ncRNA families are classified according to their secondary structures. Hence, it is important to detect a set of RNAs which have mutually similar secondary structures, because these clusters are candidates for a new ncRNA family.

We have developed a novel method which enables us to extract functional RNA clusters which have similar secondary structures from a huge set of candidates. **Figure 1** shows an outline of our method: we calculate the similarities of candidate locally stable secondary structures, construct a special graph (called a *constraint graph*) and employ a *constrained* pseudo-clique search algorithm. The method allows us to treat several *overlapping* candidates for locally stable secondary structures and this seems to be important for practical usage, because accurate prediction of RNA secondary structures using only sequences is relatively difficult [7)].

†1 Mizuho Information & Research Institute, Inc.
†2 Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology (AIST)
†3 Department of Computational Intelligence and System Science, Tokyo Institute of Technology
†4 Graduate School of Frontier Sciences, the University of Tokyo

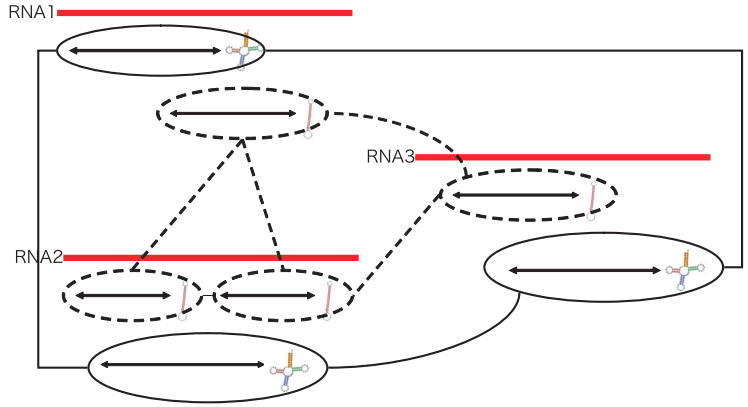

Fig. 1 Each line with left and right arrows shows a region of locally stable secondary structures in an RNA sequence. Two clusters including similar secondary structures are shown in this figure.

This paper is organized as follows. In Section 2, we describe our methods for searching for clusters which have similar secondary structures. Our problem is formulated as a kind of combinatorial optimization problem called a "constraint quasi-clique search problem" and the algorithm to solve the problem is described in Section 3. Some experimental results are reported in Section 4 and two examples of large scale sequence analysis using RNAclique are presented in Section 5. In Section 6, we discuss our results and indicate future directions for research.

2. Methods

The input of the proposed method is a set of unaligned RNA sequences $\{S_i\}_{i=1}^N$. In realistic situations, the sequences may include several functional families (whose secondary structures are similar to each other) among many unrelated (noise) sequences. Algorithm 1 gives a high-level view of our method. The details of each step in Algorithm 1 are described in the section indicated after the symbol "$\lhd$".

2.1 Calculate a Set of Candidates of Locally Stable Secondary Structures

First we calculate a set of candidates of locally stable secondary structures from input sequences $\{S_i\}_{i=1}^N$. The candidates calculated in this step may *not* be mutually exclusive structures (regions) in an RNA sequence. In other words, a structure may overlap another structure in the same sequence. Research has shown that there is a performance limitation for predicting a secondary structure from a single sequence [7),9)], because the secondary structure is formed in a complex way related to free energy, interactions with other molecules and so forth. To overcome this limitation, a comparative approach (i.e., to predict secondary structure from a set of homologous RNA sequences which form similar secondary structure to each other [9),13)]) is often used. This approach can not be used in this research because we are not able to prepare homologous RNA sequences. Instead, we provide several candidates for the secondary structure in each region using RNALfold [14)], which is an algorithm for computing locally stable RNA secondary structures with a maximal base pair span. Of course, we can also use candidates derived by other algorithms (e.g., the Rfold algorithm developed by Kiryu, et al. [18)]). In order to avoid redundant clusters and reduce computational cost, we remove similar structures in the same region from the candidate set. We do this by choosing the lower energy one, when the Matthews Correlation Coefficient (MCC) between two secondary structures is more than a given threshold α (we use $\alpha = 0.7$ in this paper). We denote by $\{s_i\}_{i=1}^M$ the candidates for locally stable secondary structures obtained in this step. For a secondary structure s_i, $seq(s_i)$ means the nucleotide sequence of the secondary structure and $seqid(s_i)$ means the sequence ID (a unique ID given to each sequence) which contains the secondary structure.

Algorithm 1 RNAclique $(\{S_i\}_{i=1}^N, maxitr, \gamma)$

Input: $\{S_i\}_{i=1}^N$, a set of RNA sequences; $maxitr$, maximum iteration of clique search; γ, minimum cluster coefficient.
1: Calculate a set of candidates of locally stable secondary structures $\{s_i\}$ $\lhd$ 2.1
2: Sequence-based filtering for $\{seq(s_i)\}$ $\lhd$ 2.2
3: Calculate similarities of secondary structures of (s_i, s_j) for all pairs passing the filter $\lhd$ 2.3
4: Construct a constraint graph $G = (V, E, c)$ $\lhd$ 2.4
5: $CS \leftarrow$ constraint_quasi_clique_search $(G, maxitr, \gamma)$ $\lhd$ 2.5
6: $CS \leftarrow$ post_process_cluster(CS) $\lhd$ 2.6
return CS

2.2 Sequence-based Filtering

We would like to compute the similarities of all pairs (s_i, s_j) of locally stable secondary structures calculated in Section 2.1, but we would have to calculate the similarities of $O(M^2)$ pairs and this would entail a huge computational cost. So sequence-based filtering is conducted before calculating the similarities. A wu-blast ⋆1 search is performed for a database $\{seq(s_i)\}_{i=1}^{N}$ using word size W and a threshold E-value E ⋆2 and we calculate the similarities (see the next section) only for the pairs which are found in this search.

2.3 Calculate Similarities between Locally Stable Structures

We calculate the similarity $sim(s_i, s_j)$ of a pair (s_i, s_j) which passes the sequenced-based filtering using the RNAforester algorithm [11] with the RIBOSUM80-65 matrix [19]. RNAforester is a tool that aligns the *secondary structure* (and sequence) of RNA molecules, and reports a score.

2.4 Construct a Constraint Graph

In this step, a special graph, called a *constraint graph*, is constructed.

Definition 1 (Constraint graph) A *constraint graph* G is an undirected graph represented as $G = (V, E, c)$ where V is a set of vertices, $E(\subset V \times V)$ is a set of edges and $c : \{S|\text{subset of } V\} \to \{0, 1\}$ is a map.

The map c generally indicates whether a subset of vertices satisfies a given condition or not. The constraint graph is a graph with constraints with respect to a subset of vertices (these conditions are described later). We construct a constraint graph whose vertex set V is given by the locally stable secondary structures calculated the method given in Section 2.1 and whose edges join two vertices whose similarity (see Section 2.3) is greater than or equal to a threshold T. We also define two types of constraint c for the graph: for $S \subset V$,

(1) $c(S) = 0$ if and only if $\exists v_1, v_2 \in S$ s.t. v_1 and v_2 belong to the same sequence (i.e., $seqid(v_1) = seqid(v_2)$).

(2) $c(S) = 0$ if and only if $\exists v_1, v_2 \in S$ s.t. the region of v_1 overlaps with that of v_2 in a sequence.

Constraint (1) is stronger than constraint (2) since $C(S) = 1$ in (1) implies $C(S) = 1$ in (2). Constraint (1) is used when searching for clusters in sequences whose length is relatively small (when we can assume there is at most one structure in one sequence), and constraint (2) is used when searching for clusters in long sequence(s) (e.g., clusters from an intronic sequence or an intergenic sequence).

2.5 Constraint Quasi-clique Search Problem

In order to detect clusters which have similar secondary structures, we would like to search a *dense* subgraph in G in this step. In order to formulate this rigorously, we make the following definitions.

Definition 2 (c-vertex set) A c-vertex set of a constraint graph $G = (V, E, c)$ is a subset $S \subset V$ which satisfies $c(S) = 1$.

Definition 3 (induced graph) For a graph $G = (V, E)$ and a subset of vertices $S \subset V$, we use G_S to denote the induced subgraph of G whose vertex set is S and whose edge set is given by $\{(v_1, v_2) \in S \times S | (v_1, v_2) \in E\}$.

Definition 4 (cluster coefficient) For a graph $G = (V, E)$, the *cluster coefficient* of G denoted by $\mathrm{cc}(G)$ is defined as $\frac{2|E|}{|V|(|V|-1)}$.

Definition 5 (γ-clique) For a graph $G = (V, E)$ and a real number $\gamma \in [0, 1]$, a subgraph $G' = (V', E')$ of G is γ-clique ⋆3 if and only if G' is a connected graph and $\mathrm{cc}(G') \geq \gamma$.

In order to detect clusters, we would like to find c-vertex sets from a constraint graph G whose induced graphs satisfy the γ-clique condition. Thus our problem can be formulated as follows.

Problem 1 Given a constraint graph $G = (V, E, c)$ and cluster coefficient γ, find c-vertices S as large as possible such that cluster coefficient of G_S is greater than or equal to γ.

This is a kind of combinatorial problem and we use the GRASP (Greedy Randomized Adaptive Search Procedure) algorithm to solve this problem by slightly modifying some previous research [1]. The algorithm is described in the next section.

For each cluster (c-vertex set) given by this step, we assign a score of multiple sequence alignment, which is calculated by the RNAforester algorithm [12].

⋆1 http://blast.wustl.edu/

⋆2 We use $E = 10$ and $W = 4$ in our experiments.

⋆3 Some say *quasi-clique* or *pseudo-clique* instead of γ-clique.

2.6 Post Processing for Clusters

After conducting constraint clique search we obtain a number of similar clusters. We reduce the number of results by post processing conducted in a greedy manner, that is, (1) order the clusters by the score, (2) select clusters one by one starting with the cluster of top rank and if the cluster overlaps another already-selected cluster and the overlap ratio is greater than δ, we discard that cluster.

3. Constraint Quasi-clique Search Algorithm

We use an approximate algorithm proposed by Abello [1], which can extract dense subgraphs from a large graph using greedy randomized adaptive search procedures (GRASP) [1]. GRASP is a general framework for solving combinatorial optimization problems [27]. In this research we modify Abello's algorithm to perform *constraint* clique search. Before describing our algorithm, we prepare some notation.

Definition 6 (Neighborhood of c-constraint vertices) Given a constraint graph (G, E, c) and c-constraint vertices S, we define a *neighborhood* of S by

$$\mathcal{N}(S) = \{y \in V \setminus S : \exists x \in S \text{ s.t. } (x, y) \in E \text{ and } c(S \cup \{y\}) = 1\} \quad (1)$$

Definition 7 (γ-Neighborhood of c-constraint vertices) Given a constraint graph (G, E, c) and c-constraint vertices S, we define a *γ-neighborhood* of S by

$$\mathcal{N}_\gamma(S) = \{y \in \mathcal{N}(S) : S \cup \{y\} \text{ is } \gamma\text{-clique}\}. \quad (2)$$

The vertex in $\mathcal{N}_\gamma(S)$ is called the *γ-vertex* of S.

Definition 8 Let $G = (V, E, c)$ be a constraint graph and $\gamma \in [0, 1]$ is given.

(1) We define the *degree* of x by $\deg(x) = |\mathcal{N}(\{x\})|$ for $x \in V$. We also define $\deg_S(x) = |\mathcal{N}(\{x\}) \cap S|$ for $x \in V$ and $S \subset V$.

(2) We define a *potential* of S by $\phi(S) = |E(G_S)| - \gamma\binom{|S|}{2}$ for $S \subset V$, where $|E(G_S)|$ is the number of the edges in G_S.

(3) For S and R with $S \cap R = \emptyset$, we define $\phi_S(R) = \phi(S \cup R)$. For $x, y \in V \setminus S$ s.t. $c(S \cup \{x, y\}) = 1$, we define $\delta_{S,x}(y) = \phi_{S\cup\{x\}}(\{y\}) - \phi_S(\{y\})$.

(4) The total effect on the potentials, caused by the selection of x is defined by

Algorithm 2 constraint_pseudo_clique_search $(G, maxitr, \gamma)$

Input: G, a constraint graph; $maxitr$, maximum number of iterations; γ, minimum cluster coefficient.
1: $CS \leftarrow \emptyset$ // Initialize a set of clusters
2: **for** $k = 1$ to $maxitr$
3: $S \leftarrow \emptyset$ // Initialize set of vertices
4: construct (S, G, γ, p_1, p_2, p_3) // construct initial vertex set
5: localSearch (S, G, γ) // improve vertex set
6: $CS \leftarrow CS \cup \{S\}$
7: **endfor**
8: remove duplicate clusters from CS
9: **return** CS

$$\begin{aligned}\Delta_S(x) &= \sum_{y \in \{y' \in \mathcal{N}_\gamma(S) : c(S\cup\{x,y\})=1\}} \delta_{S,x}(y) \\ &= |\mathcal{N}_\gamma(\{x\})| + |\mathcal{N}_\gamma(S)|\left(\deg_S(x) - \gamma(|S|+1)\right).\end{aligned}$$

A high level description of the iterative search for γ-cliques is given as Algorithm 2. In the first step, we construct γ-cliques in a greedy manner (line 4) and improve the γ-clique by a kind of local search technique (line 5). The result is that we obtain one cluster at each iteration and get at most $maxitr$ clusters (because we may get the same cluster).

In the construction step (see Algorithm 3), we greedily select vertices from the neighborhood (see Definition 6 and Definition 7) in line 6 and line 8. The greedy selection in Algorithm 3 is described as Algorithm 4. In each loop, starting at line 3 in Algorithm 3, one vertex is added to the current vertex set. We select the vertex which has a high potential as defined in Definition 8 part (4). Note that the parameter p in Algorithm 4 indicates the randomness of the selection: we randomly select the vertex if $p = 0$.

In the local search step (see Algorithm 5), we improve the γ-clique obtained by the construction step. In this local search, our search space is limited to the graphs given by deleting one vertex and adding two vertices to the current graph (line 5).

4. Experiments

In this section, we confirm the effectiveness of the proposed method by computational experiments. All experiments in this and the next section were con-

Algorithm 3 construct $(S, G, \gamma, p_1, p_2, p_3)$

Input: G, constraint graph; γ, threshold of cluster coefficient; $p_k \in [0,1](k = 1,2,3)$.
1: $\gamma^* \leftarrow 1$
2: $S^* \leftarrow$ getGRASPSelection $(\deg, p_1)$
3: **while** $\gamma^* \geq \gamma$
4: $S \leftarrow S^*$
5: **if** $\mathcal{N}_{\gamma^*}(S) \neq \emptyset$ **then**
6: $x \leftarrow$ getGRASPSelection $(\mathcal{N}_{\gamma^*}(S), \Delta_S, p_2)$
7: **elseif** $\mathcal{N}(S) \neq \emptyset$ **then**
8: $x \leftarrow$ getGRASPSelection $(\mathcal{N}(S), \deg_S, p_3)$
9: **else**
10: **return** S
11: **endif**
12: $S^* \leftarrow S \cup \{x\}$
13: $\gamma^* \leftarrow |E(G_{S^*})| / \binom{|S^*|}{2}$
14: **endwhile**
15: **return** S

Algorithm 4 getGRASPSelection (C, f, p)

Input: C. candidate set; f, score function; p, parameter that determines search space.
1: $m \leftarrow \min\{f(t) | t \in C\}$
2: $M \leftarrow \max\{f(t) | t \in C\}$
3: $RCL \leftarrow \{y \in C | f(y) \geq m + p(M - m)\}$
4: Select s at random from RCL
5: **return** s

Algorithm 5 localSearch (S, G, γ)

Input: S, current c-vertex set; G, constraint graph; γ, minimum cluster coefficient.
1: $\mathcal{N} \leftarrow \{S' : S' = (S \setminus \{x\}) \cup \{y\} \cup \{z\}$ and $G_{S'}$ is connected and $\mathrm{cc}(G_{S'}) \geq \gamma\}$
2: **while** $|\mathcal{N}| > 0$
3: select $S' \in \mathcal{N}(S)$
4: $S \leftarrow S'$
5: $\mathcal{N} \leftarrow \{S' : S' = (S \setminus \{x\}) \cup \{y\} \cup \{z\}$ and $G_{S'}$ is connected and $\mathrm{cc}(G_{S'}) \geq \gamma\}$
6: **endwhile**
7: **return** S

ducted using cluster machines, each of which has two 2 GHz processors (AMD Opteron(tm) Processor 246) and 4 Gbyte memory.

4.1 Experiments Using Our Own Datasets

In this section, we present the experiments using datasets selected by us from the Rfam database.

Table 1 ncRNA families in our dataset. "#seqs" means the number of sequences in a family, "ave len" means the average length of sequence and "ident" means the averaged pairwise identities.

Family	#seqs	ave len	ident
Flavi_CRE	20	96	59
Flavivirus_DB	20	72.1	71
Hammerhead_1	20	64.1	61
Hammerhead_3	20	55.8	69
Purine	20	99.6	54
S-element	12	68.2	76
SNORD113	20	75.3	70
SRP_bact	20	92.9	50
Tombus_3_III	20	69.9	80
Trp_leader	11	98	61
UnaL2	20	54	78
Vimentin3	19	68.9	72
Y	16	94.7	63
ctRNA_p42d	14	76.5	67
ctRNA_pGA1	15	77.7	66
ctRNA_pT181	16	96.2	77
tRNA	20	72.6	42

4.1.1 Datasets

From Rfam version 8.0 (February 2007, 574 families) ⋆1 we choose 17 families that satisfy all of the following conditions: (1) the common secondary structure is confirmed by previous papers, (2) the number of sequences in the family is ≥ 10, (3) the average length is < 100 and > 50, (4) the average pairwise identity is $< 80\%$ and (5) they do not contain pseudoknotted structures. If a selected family contains more than 20 sequences, we randomly choose 20 sequences from the family. **Table 1** shows the details of selected families. From those families, we created two datasets, **Dataset1** and **Dataset2**. Each sequence in **Dataset1** is created by putting a sequence in Table 1 between two random sequences of length 500. **Dataset2** contains one long sequence constructed by connecting all sequences in Table 1 with random sequences of length 500. (See **Fig. 2** for schematic illustrations of the two datasets.) The details of the datasets are described in **Table 2**.

⋆1 http://www.sanger.ac.uk/Software/Rfam/

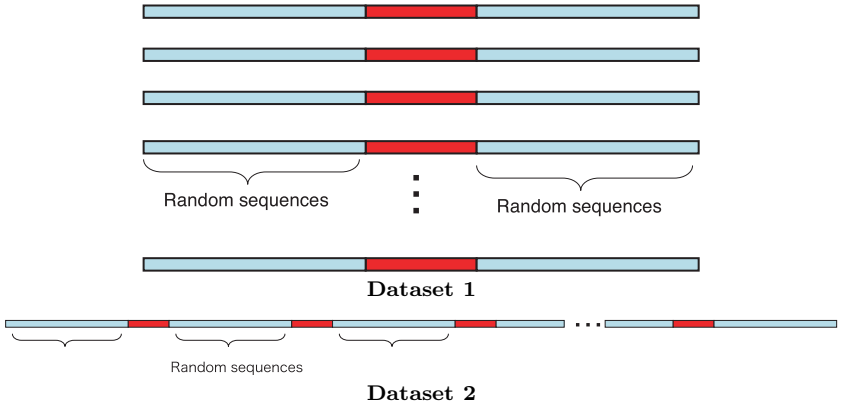

Fig. 2 Schematic views of **Dataset1** and **Dataset2**. The red regions are RNA sequences in Table 1 and the light blue regions are random sequences of 500 bases. Each dataset contains all of RNA sequences in Table 1.

Table 2 Summary of the datasets using in our experiments. Each dataset includes all sequences in Table 1. See also Fig. 2.

	Dataset1	**Dataset2**
Number of sequences	303	1
Total # residues	326543	175543
Smallest	1043	175543
Largest	1118	175543
Average length	1078	175543

4.1.2 Performance Measures

For clustering a segmented dataset, Will, et al. [33] have proposed ROC evaluation. However, this evaluation can not be applied here because our method does not conduct clustering and does not produce mutually exclusive clusters. Therefore, we consider several evaluation measures inspired by standard evaluation measures using in bi-clustering experiments.

First, we determine whether a local predicted region matches a motif region or not. For a motif region r' and predicted region r in a sequence, TP is the number of nucleotides which are in both the motif and the predicted region, FP is the number of nucleotides which are in the predicted region but not in the reference region, FN is the number of nucleotides which are in the reference region but not in the predicted region and TN is the number of nucleotides which are not in either the reference or the predicted region. Then we define the sensitivity (Sen), specificity (Spe) and accuracy (Acc) of nucleotides as follows:

$$\mathrm{Sen}(r,r') = \frac{TP}{TP+FN},\ \mathrm{Spe}(r,r') = \frac{TP}{TP+FP},$$
$$\mathrm{Acc}(r,r') = \sqrt{\mathrm{Sen}(r,r') \times \mathrm{Spe}(r,r')}.$$

In this case, $\mathrm{Acc}(r,r')$ is a good approximation of the Matthews correlation coefficient (MCC)

$$\mathrm{MCC}(r,r') = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

because $TN >> TP, FP, TN$. For a threshold β_1, we say the predicted region r *matches* to the motif region r' if $\mathrm{Acc}(r,r') \geq \beta_1$ (we use $\beta_1 = 0.5$ in our experiments).

Next we define measures which check the quality of a predicted set of clusters. (Note that our method produces a set of clusters in a single calculation.) Given the optimal set of clusters $\mathcal{C}_{opt}$ and a set of clusters $\mathcal{C}$, we define two measures for $\mathcal{C}$:

$$\mathrm{Relevance}(\mathcal{C}) = S(\mathcal{C}, \mathcal{C}_{opt}) \text{ and } \mathrm{Recovery}(\mathcal{C}) = S(\mathcal{C}_{opt}, \mathcal{C})$$

where

$$S(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1|} \sum_{C_1 \in \mathcal{C}_1} \max_{C_2 \in \mathcal{C}_2} \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}.$$

These two measures are often used for evaluating bi-clustering methods e.g., [25]. We define another relevance measure by

$$\mathrm{Relevance2}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \max_{C_{opt} \in \mathcal{C}_{opt}} \frac{|C \cap C_{opt}|}{|C|}.$$

Relevance2 reflects the average specificity of clusters. In other words, if each predicted cluster has only a unique family, Relevance2 for predicted clusters is near 1.

Table 3 Results for dataset1 and dataset2. Method 1, Method 2 and Method 3 refer to the modified methods described in Section 4.1.3. Rel and Rec means Relevance and Recovery, respectively.

	Proposed		Method 1		Method 2		Method 3	
	Rel	Rec	Rel	Rec	Rel	Rec	Rel	Rec
Dataset1	0.711	0.846	0.201	0.301	0.711	0.687	0.331	0.500
Dataset2	0.716	0.837	0.307	0.362	0.722	0.822	0.350	0.525

4.1.3 Comparison Methods

We now compare our method with three methods which are slight modifications of the proposed method.

Modified method 1 Remove conflict structures greedily on step 1 in Algorithm 1 (i.e., choose the lower energy structure if two structures overlap).

Modified method 2 Use a sequence similarity measure instead of a measure which considers both sequence and secondary structures. We use a score of clustalW for sequence similarity.

Modified method 3 We search for a clique subgraph (γ-clique for $\gamma = 1$) instead of a quasi-clique subgraph.

4.1.4 Results

We used constraint (1) for Dataset1 and constraint (2) for Dataset2. The results for using the four methods on **Dataset1** and **Dataset2** are shown in **Table 3**. The performance of the proposed method is better than that of the modified methods. In particular, the proposed method is much better than Method 1, which means that it is important to allow overlapped secondary structures. The proposed method is also much better than Method 3, which indicates that it is useful to search quasi-cliques instead of cliques. Moreover, by comparing the proposed method and Method 2, we see that using a similarity measure which considers both sequence and secondary structure produces better Recovery than using a similarity measure which only considers sequence.

4.2 Experiment Using Dataset of Will, et al.

In this experiment, we use the dataset from Will, et al. [33]. Will, et al. conducted a hierarchical clustering for seed sequences in the Rfam database [8]. In contrast to the datasets used in the previous experiment, each sequence is a "segmented" sequence (i.e., each sequence is a complete segment of RNA genes).

Table 4 Results of proposed method (RNAclique) for the dataset of Will, et al. [33]

#clusters	Relevance 2	Relevance	Recovery
828	0.878	0.373	0.637

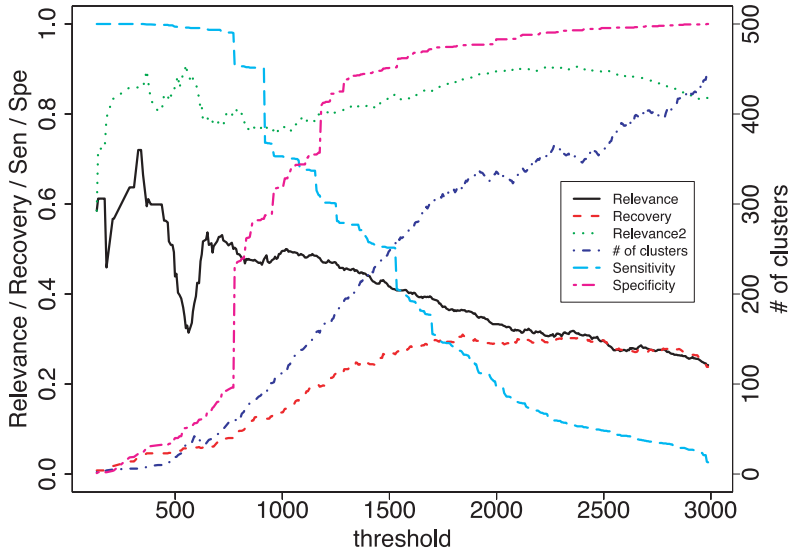

Fig. 3 Performance using LocARNA pipeline [33]. The x-axis indicates a threshold for obtaining a set of clusters from the dendrogram. The y-axis indicates Relevance, Recovery, Sensitivity, Specificity or the number of clusters.

The results of the proposed method for this dataset are shown in **Table 4** and the performance of the LocARNA pipeline [33] for obtaining clusters from the dendrogram is shown in **Fig. 3**. It can be seen that the Relevance measures of the two methods are not so good. However, Relevance2 for each cluster in the set of predicted clusters is more than 0.85, and this means that most of the sequences in each cluster are contained in one family. In particular, the Recovery of RNAclique is much better than that using the method of Will, et al.

5. Two Examples of Large Scale Sequence Analysis

In this section, we present two examples of large scale similarity search using

RNAclique.

5.1 Detecting miRNA clusters in a long intergenic sequence

There are a number of RNA families which form a cluster at a narrow region in a genomic sequence. For example, the region which starts at 58,861,740 and ends at 58,957,496 of chromosome 19 in human genome (hg17) contains 43 miRNAs. In this experiment, the input is the long sub-sequences of chromosome 19 (repeated region is masked) and the constraint graph is constructed using constraint (2).

Using RNAclique, we detected 55 clusters and the rank 1 cluster of the results contains 54 sequences including 43 miRNAs and 1 miRNA candidate predicted by Berezikov, et al. [3], while there is no annotation related to miRNA for the others. The calculation time was 951 seconds using 16CPUs (4.2 hours if we had used one CPU).

5.2 Detecting Clusters in ncRNA Candidates Predicted by Washietl, et al.

Washietl, et al. predicted about 35,000 ncRNA candidates from the human genome [31] using RNAz [32]. In this experiment, our input is all of those sequences (including both plus and minus strands) and we use constraint (1) for constructing the constraint graph from the dataset. The total number of sequences in the input data is 71,970 and the average length of those is 152 bases (the smallest is 51 bases and the largest is 1,320 bases). Using RNAclique, 492 clusters were detected in this experiment. We collect the known RNA sequences in Rfam database, which are mapped to the human genome and overlap with the region predicted by Washietl, et al. [31] (the overlap ratio is at least 80%). Using these data, we investigated how RNAclique recovers the known families, and the best coverage of clusters for each family is shown in **Table 5**. The calculation time was 64,708 seconds using 28CPUs (21 days if we had used 1 CPU). We find that most of the top-ranked clusters contain sequences with no annotation.

Table 5 Coverage of known families in the predictions of RNAclique. "Coverage" means the best coverage of cluster among all predictions.

Family	Coverage
SECIS	0.14 (1 / 7)
U70	0.84 (16 / 19)
let-7	0.82 (9 / 11)
mir-10	0.80 (4 / 5)
tRNA	0.31 (13 / 42)

6. Discussion and Future Directions

In this paper we have proposed a novel method which enables us to search RNA sequences for clusters containing locally stable secondary structures which are similar to each other. Using a constraint quasi-clique search algorithm, the method allows us to treat several overlapped structures as candidates for secondary structures, and we can also add candidate structures predicted by other algorithms (e.g., Kiryu, et al. [18]) to our method. Note that each step of our method can be easily parallelized, and this enables us to apply our method to a dataset including a large number of sequences (we presented two examples of large scale sequence analysis using RNAclique in Section 5). We are currently working on applying our method to large scale sequence data. We will apply RNAclique to each intron of protein-coding genes (in order to discover new ncRNA family clusters in intron regions) and the set of continuous conserved regions in the human genome.

One drawback of our method is in the detection of the number of clusters in general. In order to obtain more confident clusters, we have developed a method for finding clusters containing locally stable secondary structures that not only have similar structures to each other but also have other features in common, e.g., evidence of expression (a support of a tiling array or ESTs), or overlap with an interesting region (ultraconserved elements [2], human accelerated regions [28], indel-conserved regions [20], transposon free regions [29] and so forth). These useful features are summarized in a database developed by Mituyama, et al. [17),23] (Also see http://www.ncrna.org).

Acknowledgments This work was partially supported by the "Functional RNA Project" funded by the New Energy and Industrial Technology Development Organization (NEDO) of Japan, and was partially supported by a Grant-in-Aid for Scientific Research on Priority Areas "Comparative Genomics" from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

The authors are grateful to our colleagues at the Computational Biology Research Center (CBRC) for fruitful discussions. Also, the constructive comments of the anonymous reviewers are greatly appreciated.

References

1) Abello, J., Resende, M.G.C. and Sudarsky, S.: Massive quasi-clique detection, *LATIN '02: Proc. 5th Latin American Symposium on Theoretical Informatics*, pp.598–612, London, UK, Springer-Verlag (2002).
2) Bejerano, G., Pheasant, M., Makunin, I., Stephen, S., Kent, W.J., Mattick, J.S. and Haussler, D.: Ultraconserved elements in the human genome, *Science*, Vol.304, No.5675, pp.1321–1325 (May 2004).
3) Berezikov, E., Guryev, V., van de Belt, J., Wienholds, E., Plasterk, R. and Cuppen, E.: Phylogenetic shadowing and computational identification of human microRNA genes, *Cell*, No.120, pp.21–24 (Jan. 2005).
4) Carninci, P. and Hayashizaki, Y.: Noncoding RNA transcription beyond annotated genes, *Curr. Opin. Genet. Dev.*, No.17, pp.139–144 (Apr. 2007).
5) Costa, F.F.: Non-coding RNAs: New players in eukaryotic biology, *Gene*, Vol.357, No.2, pp.83–94 (Sep. 2005).
6) Eddy, S.R.: Non-coding RNA genes and the modern RNA world, *Nat. Rev. Genet.*, Vol.2, No.12, pp.919–929 (Dec. 2001).
7) Gardner, P.P. and Giegerich, R.: A comprehensive comparison of comparative RNA structure prediction approaches, *BMC Bioinformatics*, No.5, p.140 (Sep. 2004).
8) Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S.R. and Bateman, A.: Rfam: Annotating non-coding RNAs in complete genomes, *Nucleic Acids Res.*, 33 (Database issue), pp.121–124 (Jan. 2005).
9) Hamada, M., Kiryu, H., Sato, K., Mituyama, T. and Asai, K.: Prediction of RNA secondary structure using generalized centroid estimators, *Bioinformatics Advance Access published on December 18* (2008).
10) Hamada, M., Tsuda, K., Kudo, T., Kin, T. and Asai, K.: Mining frequent stem patterns from unaligned RNA sequences, *Bioinformatics*, No.22, pp.2480–2487 (Oct. 2006).
11) Hōchsmann, M., Tōller, T., Giegerich, R. and Kurtz, S.: Local similarity in rna secondary structures, *CSB '03: Proc. IEEE Computer Society Conference on Bioinformatics*, p.159, Washington, DC, USA, IEEE Computer Society (2003).
12) Höchsmann, M., Voss, B. and Giegerich, R.: Pure multiple rna secondary structure alignments: A progressive profile approach, *IEEE/ACM Trans. Comput. Biology Bioinform.*, Vol.1, No.1, pp.53–62 (2004).
13) Hofacker, I.L., Fekete, M. and Stadler, P.F.: Secondary structure prediction for aligned RNA sequences, *J. Mol. Biol.*, Vol.319, No.5, pp.1059–1066 (June 2002).
14) Hofacker, I.L., Priwitzer, B. and Stadler, P.F.: Prediction of locally stable RNA secondary structures for genome-wide surveys, *Bioinformatics*, Vol.20, No.2, pp.186–190 (Jan. 2004).
15) Huttenhofer, A., Schattner, P. and Polacek, N.: Non-coding RNAs: hope or hype?, *Trends Genet.*, Vol.21, No.5, pp.289–297 (May 2005).
16) Kapranov, P., Cheng, J., Dike, S., Nix, D.A., Duttagupta, R., Willingham, A.T., Stadler, P.F., Hertel, J., Hackermuller, J., Hofacker, I.L., Bell, I., Cheung, E., Drenkow, J., Dumais, E., Patel, S., Helt, G., Ganesh, M., Ghosh, S., Piccolboni, A., Sementchenko, V., Tammana, H. and Gingeras, T.R.: RNA maps reveal new RNA classes and a possible function for pervasive transcription, *Science*, Vol.316, No.5830, pp.1484–1488 (June 2007).
17) Kin, T., Yamada, K., Terai, G., Okida, H., Yoshinari, Y., Ono, Y., Kojima, A., Kimura, Y., Komori, T. and Asai, K.: fRNAdb: A platform for mining/annotating functional RNA candidates from non-coding RNA sequences, *Nucleic Acids Res.*, No.35 (Database issue), pp.145–148 (Jan. 2007).
18) Kiryu, H., Kin, T. and Asai, K.: Rfold: An exact algorithm for computing local base pairing probabilities, *Bioinformatics*, No.24, pp.367–373 (Feb. 2008).
19) Klein, R.J. and Eddy, S.R.: RSEARCH: Finding homologs of single structured RNA sequences, *BMC Bioinformatics*, No.4, p.44 (Sep. 2003).
20) Lunter, G., Ponting, C.P. and Hein, J.: Genome-wide identification of human functional DNA using a neutral indel model, *PLoS Comput. Biol.*, Vol.2, No.1, e5 (Jan. 2006).
21) Matera, A.G., Terns, R.M. and Terns, M.P.: Non-coding RNAs: Lessons from the small nuclear and small nucleolar RNAs, *Nat. Rev. Mol. Cell. Biol.*, Vol.8, No.3, pp.209–220 (Mar. 2007).
22) Mattick, J.S. and Makunin, I.V.: Non-coding RNA, *Hum. Mol. Genet.*, 15 Spec No.1, pp.17–29 (Apr. 2006).
23) Mituyama, T., Yamada, K., Hattori, E., Okida, H., Ono, Y., Terai, G., Yoshizawa, A., Komori, T. and Asai, K.: The Functional RNA Database 3.0: Databases to support mining and annotation of functional RNAs, *Nucleic Acids Res.* (Oct. 2008).
24) Nakaya, H.I., Amaral, P.P., Louro, R., Lopes, A., Fachel, A.A., Moreira, Y.B., El-Jundi, T.A., da Silva, A.M., Reis, E.M. and Verjovski-Almeida, S.: Genome mapping and expression analyses of human intronic noncoding RNAs reveal tissue-specific patterns and enrichment in genes related to regulation of transcription, *Genome Biol.*, Vol.8, No.3, R43 (2007).
25) Okada, Y., Fujibuchi, W. and Horton, P.: A biclustering method for gene expression module discovery using closed itemset enumeration algorithm, *IPSJ Transactions on Bioinformatics*, No.48 (SIG5), pp.39–48 (2007).
26) Pedersen, J.S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E.S., Kent, J., Miller, W. and Haussler, D.: Identification and classification of conserved RNA secondary structures in the human genome, *PLoS Comput. Biol.*, Vol.2, No.4, e33 (Apr. 2006).
27) Pitsoulis, L. and Resende, M.: Greedy randomized adaptive search procedures, Pardalos, P. and Resende, M. (Eds.), *Handbook of Applied Optimization*, pp.168–183. Oxford University Press (2002).

28) Pollard, K.S., Salama, S.R., Lambert, N., Lambot, M.-A., Coppens, S., Pedersen, J.S., Katzman, S., King, B., Onodera, C., Siepel, A., Kern, A.D., Dehay, C., Igel, H., Ares, M.J., Vanderhaeghen, P. and Haussler, D.: An RNA gene expressed during cortical development evolved rapidly in humans, *Nature*, Vol.443, No.7108, pp.167–172 (Sep. 2006).
29) Simons, C., Pheasant, M., Makunin, I.V. and Mattick, J.S.: Transposon-free regions in mammalian genomes, *Genome Res.*, Vol.16, No.2, pp.164–172 (Feb. 2006).
30) Torarinsson, E., Sawera, M., Havgaard, J.H., Fredholm, M. and Gorodkin, J.: Thousands of corresponding human and mouse genomic regions unalignable in primary sequence contain common RNA structure, *Genome Res.*, Vol.16, No.7, pp.885–889 (July 2006).
31) Washietl, S., Hofacker, I.L., Lukasser, M., Huttenhofer, A. and Stadler, P.F.: Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome, *Nat. Biotechnol.*, Vol.23, No.11, pp.1383–1390 (Nov. 2005).
32) Washietl, S., Hofacker, I.L. and Stadler, P.F.: Fast and reliable prediction of noncoding RNAs, *Proc. Natl. Acad. Sci. USA*, Vol.102, No.7, pp.2454–2459 (Feb. 2005).
33) Will, S., Reiche, K., Hofacker, I., Stadler, P. and Backofen, R.: Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering, *PLoS Comput. Biol.*, No.3, e65 (Apr. 2007).
34) Zhang, Z., Pang, A.W.C. and Gerstein, M.: Comparative analysis of genome tiling array data reveals many novel primate-specific functional RNAs in human, *BMC Evol. Biol.*, No.7 (Suppl) 1, S14 (2007).

(Received October 27, 2008)
(Accepted December 14, 2008)
(Released March 24, 2009)

(Communicated by *Tatsuya Akutsu*)

Michiaki Hamada was born in 1977. He received his M.S. from Tohoku University in 2002, and will receive Ph.D. from Tokyo Institute of Technology at March in 2009. He has been working at Mizuho Information Research & Institute, Inc. (which was previously called Fuji Research Institute Corporation) since 2002. His current interests are application of data mining or machine learning techniques to biological data including RNAs. He has attended "Functional RNA Project" funded by NEDO since 2005, and been studying sequence analysis technologies for functional RNAs. He received the Aoba-Rigaku Award and the Tohoku Univ. Math. Award in 2000.

Toutai Mituyama was born in 1968. He received his masters degree and Ph.D. from Japan Advanced Institute of Science and Technology (JAIST) in 2000 and 2003, respectively. He has been working, as a research scientist, for Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST). His current research interests are noncoding RNA finding and database development. He is currently committing to the Functional RNA Project funded by New Energy and Technology Development Organization (NEDO) since 2005. He is a member of the Japanese Society for Bioinformatics, the RNA Society of Japan and the Molecular Biology Society of Japan.

Kiyoshi Asai was born in 1960. He received his M.S. and Ph.D. from the University of Tokyo in 1985 and 1995, respectively. He has been a professor of Department of Computational Biology, the University of Tokyo since 2003, and has been a director of Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST) since 2007. His current interests are mathematical models for biological sequence analysis and comparative genome analysis. He is currently committing to the Functional RNA Project funded by NEDO as a group leader of Bioinformatics group. He is a member of the Japanese society for Bioinformatics, International Society for Computational Biology (ISCB) and Information Processing Society of Japan.