Maximum Common Substructure Extraction in RNA Secondary Structures Using Clique Detection Approach

Shih-Yi Chao

Open Science Index, Computer and Information Engineering Vol:2, No:9, 2008 publications.waset.org/9393.pdf

Abstract**—**The similarity comparison of RNA secondary structures is important in studying the functions of RNAs. In recent years, most existing tools represent the secondary structures by tree-based presentation and calculate the similarity by tree alignment distance. Different to previous approaches, we propose a new method based on maximum clique detection algorithm to extract the maximum common structural elements in compared RNA secondary structures. A new graph-based similarity measurement and maximum common subgraph detection procedures for comparing purely RNA secondary structures is introduced. Given two RNA secondary structures, the proposed algorithm consists of a process to determine the score of the structural similarity, followed by comparing vertices labelling, the labelled edges and the exact degree of each vertex. The proposed algorithm also consists of a process to extract the common structural elements between compared secondary structures based on a proposed maximum clique detection of the problem. This graph-based model also can work with NC-IUB code to perform the pattern-based searching. Therefore, it can be used to identify functional RNA motifs from database or to extract common substructures between complex RNA secondary structures. We have proved the performance of this proposed algorithm by experimental results. It provides a new idea of comparing RNA secondary structures. This tool is helpful to those who are interested in structural bioinformatics.

Keywords**—**Clique detection**,** labeled vertices, RNA secondary structures, subgraph, similarity.

I. Introduction

COMPARING the similarities of RNA secondary structures is one of the challenging tasks in molecular biology. Similar structures often imply similar functions. If there is a newly discovered RNA, a suggestion of its function can be obtained by comparing its similarity to known RNA structures by finding the relatively common structures of known RNAs and by measuring the overall similarities among the whole RNA secondary structures. It is known that if there is an amount of RNA conserved on the primary sequence level when performing a primary sequence alignment, it may have a similar function, and may fold conserved secondary structural regions among these sequences. However, functional RNA families, such as tRNA and rRNA, are highly conserved on secondary structures but little on primary sequences. Therefore, it is useful to compare RNA secondary structures directly without accomplishing primary sequence alignment. We propose a graph-based approach for purely comparing RNA secondary structures, which not only calculates the similarity scores between compared RNA secondary structures, but also detects the maximum common secondary structures on the basis of maximum clique detection; that is, the detection of the maximum common subgraph between two graphs after transforming secondary structures into simple graphs.

Most existing studies, such as [1][2][3][4][5][6], they transform the RNA secondary structures into node-labeled, tree-like structures based on the assumption that any base exists at most in one such pair and the edges of the bonded pairs are non-crossing. Some tools, such as Vienna's RNAdistance [7], use a tree-liked model to represent RNA secondary structures and compare these structures on the basis of edit distance. The Vienna's RNAforester tool [7], as its name suggests, extends the tree-like model to a forest-like model, which significantly improves time complexities. Another RNA secondary structure searching algorithm provided by Macke *et al.* [8] is based on the motif definition of unique structural and functional properties. They develop a program that can describe the RNA structural motifs and then search any nucleotide sequence databases.

In this approach, based on detecting the largest clique of the maximum common subgraph problem is presented, it can be used in similar structural elements extracting applications where secondary structures are presented as simple graphs so that the nucleotides correspond to labeled vertices and the helixes correspond to labeled edges. In addition, we also develop a measurement method that provides scores to indicate the similarity of the structures compared. In other words, this proposed algorithm is composed of two major sections. The first provides the similarity score of the secondary structures compared, and the second finds the maximum common subgraph of two graphs that are transformed by the RNA secondary structures. Contrasting this proposed algorithm with previous related research, there are three major contributions. The first one is that, in the similarity measurement section, we consider not only the diverse number of vertices and edges represented in the two compared graphs, but also take account of vertices and edge labelling, which furnish more discriminating similarity scores. The second major contribution

SY. C. Author is with the Computer Science and Information Engineering Department, University of Ching Yun, Jung Li, 229 Taiwan (e-mail: chaosy@cyu.edu.tw; phone: 886-3-4581196 ext. 7720).

is the detection of the largest cliques from the compatibility graph to obtain the maximum common subgraph, which is constructed from two compared simple graphs. The third contribution is the proposed algorithm can work with the NC-IUB code (the nucleotide abbreviations recommended by the Nomenclature Committee of the International Union of Biochemistry), and search for similar structural elements between compared structures.

II. Methods

A. Data Sets

A human UTR structure database was constructed as follows. We download human RefSeq mRNA sequences (January 2007 version) from NCBI [9]. Each RefSeq sequence is processed to extract the 5'UTR and 3'UTR sequences, and a 50nt sliding window is moved along the input sequence with a step length of 10nt. The RNAFold program generates secondary structures and the lowest free energy is kept for each folding sequence. To test the performance of the proposed algorithm on complex secondary structures, we also download the 5S ribosomal RNA (5S rRNA) family sequences (X02979 and X07545), U12 minor spliceosomal RNA sequences for human (L43846.1) and mouse (L43843.1), and let-7 precursor microRNA sequences for *C.elegans* and human from the RFAM database.

B. Definitions

A graph G consists of a set of vertices $V(G)$ and a set of edges $E(G)$. In a simple graph, two of the vertices in G are linked if there exists an edge $(v_i, v_j)\in E(G)$ connecting the vertices v_i and v_j in graph G such that $v_i\in V(G)$ and $v_j\in V(G)$. The number of vertices will be denoted by $|V(G)|$, and the set of vertices adjacent to a vertex v_i is referred to as the neighbors of v_i, $N(v_i)$. The degree of a vertex v_i is the number of edges with which it is incident, symbolized by $d(v_i)$. Two graphs, G_1 and G_2, are said to be isomorphic ($G_1 \cong G_2$) if a one-to-one transformation of V_1 onto V_2 effects a one-to-one transformation of E_1 onto E_2. A subgraph G' of a graph G is a graph whose set of vertices and set of edges satisfy the relations: $V(G')\subseteq V(G)$ and $E(G') \subseteq E(G)$, and if G' is a subgraph of G, then G is said to be a supergraph of G'. The line graph $L(G)$ of an undirected graph G is a graph such that each vertex in $L(G)$ indicates an edge in G and any pairs of vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G.

C. The Measurement of Structural Similarity

Before the similarity of RNA secondary structures can be measured, we transform the combinations of parentheses and dots that RNAFold output into a simple graph, which is demonstrated in Figs. 1(a), 1(b) and 1(c). The proposed similarity measure procedure is based on the divergence of the vertices and edges displayed in G_1 and G_2. The degree sequence of a graph is the list of vertex degrees, usually written in non-increasing order. The vertices in each graph are separated into four partitions by the nucleotide types, and we sort these vertices according to the degrees by descending order, which means we have the degree sequence of every partition for the compared graphs G_1 and G_2. Let $P^1{}_i$ and $P^2{}_i$ as given in (1) and (2) denote the sequences of sorted vertices in partition i in graphs G_1 and G_2, respectively.

$$p_i^1 = \{v \in V(G_1) : Label\ (v) = i\} \tag{1}$$

$$p_i^2 = \{v \in V(G_2) : Label\ (v) = i\} \tag{2}$$

where i indicates the number of partitions, and the *Label* denotes the distinct types of nucleotides. Therefore, the vertex similarity between a pair of graphs can be given as follows:

$$S_V(G_1, G_2) = \sum_{i=1}^{l} \min \{|\ p_i^1\ |, |\ p_i^2\ |\} \tag{3}$$

Instead of only considering the degree of each vertex in partitions, we also contemplate the nucleotide types of adjacent vertices. In other words, an accumulation is increased to a pair of compared vertices in G_1 and G_2, by who's each edge incident to the neighbour vertex with the same content of nucleotides, which is denoted as the function $f_N(p_i^1, p_i^2)$, where p_i^1 and p_i^2 represent the partitions in the compared graphs, as described above.

$$\zeta_N(G_1,G_2) = \left\lfloor \frac{1}{2}\sum_{i=1}^{l} f_N(p_i^1, p_i^2) \right\rfloor \tag{4}$$

$$f_N(p_i^1, p_i^2) = \sum_{j=1}^{k} \left(\sum_{v_{jq}^1 \in N(v_j^1),\ v_{jq}^2 \in N(v_j^2)} \sigma_C(v_{jq}^1, v_{jq}^2) \right), \qquad k = \max\{|\ p_i^1\ |, |\ p_i^2\ |\} \tag{5}$$

$$\sigma_C(v_{jq}^1, v_{jq}^2) = \begin{cases} 1, Content(v_{jq}^1) = Content(v_{jq}^2) \\ 0, Content(v_{jq}^1) \neq Content(v_{jq}^2) \end{cases} \tag{6}$$

Note that the term on the right of Eq. (4) is divided by two since each edge in this formulation is counted twice, and the *Content* described in Eq. (6) denotes the contents (A, U, C or G) of nucleotides. If the content of nucleotide of v_{jq}^1 is equal to v_{jq}^2, the result of $\sigma_C(v_{jq}^1, v_{jq}^2)$ will be 1. Moreover, we also consider three kinds of distinct edge types, L, N, and S, represented as edges in the loop, in the pairing, and in the stem, respectively. As a result, the type of incident edges of each vertex in a pair of compared graphs is denoted as $e^1{}_j$ and $e^2{}_j$ for G_1 and G_2, respectively. The function, $f_T(p_i^1, p_i^2)$, is defined to be a linear assignment of consistent edge types associated with each pair of vertices in the sequence p_i^1 and p_i^2 such that each vertex in p_i^1 is compared to each vertex in p_i^2. The Type shown in Eq. (11) indicates the types of edges, and if the edge type of e_{jq}^1 is equal to e_{jq}^2, then the value of $\sigma_T(e_{jq}^1, e_{jq}^2)$ will be 1.

$$\mathbf{e}_j^1 = \{ e \in (v_j^1, N(v_j^1)) \} \tag{7}$$

$$\mathbf{e}_j^2 = \{ e \in (v_j^2, N(v_j^2)) \} \tag{8}$$

$$\zeta_T(G_1,G_2) = \left\lfloor \frac{1}{2}\sum_{i=1}^{l} f_T(p_i^1, p_i^2) \right\rfloor \tag{9}$$

$$f_T(p_j^1, p_j^2) = \sum_{j=1}^{k}\left(\sum_{e_{jq}^1 \in \mathbf{e}_j^1,\, e_{jq}^2 \in \mathbf{e}_j^2} \sigma_T(e_{jq}^1, e_{jq}^2)\right), k = \max\{|\, p_i^1\,|, |\, p_i^2\,|\} \quad (10)$$

$$\sigma_T(e_{jq}^1, e_{jq}^2) = \begin{cases} 1, Type(e_{jq}^1) = Type(e_{jq}^2) \\ 0, Type(e_{jq}^1) \neq Type(e_{jq}^2) \end{cases} \quad (11)$$

With $\zeta_N(G_1, G_2)$ and $\zeta_T(G_1, G_2)$ being calculated as previously discussed, the edge similarity between two given graphs can be given as follows:

$$S_E(G_1, G_2) = \alpha \cdot \zeta_N(G_1, G_2) + \beta \cdot \zeta_T(G_1, G_2) \quad (12)$$

where the parameters α and β indicate the significance of $\zeta_N(G_1, G_2)$ and $\zeta_T(G_1, G_2)$. Since the resemblance to the vertex and edge of the two compared graphs are estimated, a similarity measure between a pair of graphs can be given as follows:

$$Sim(G_1\,, G_2) = \frac{(\,S_V(G_1\,, G_2) + S_E(G_1\,, G_2))^2}{(\,|V(G_1)| + |E(G_1)|\,) \cdot (\,|V(G_2)| + |E(G_2)|\,)} \quad (13)$$

D. Encode the Graphs

The RNA secondary structure may consist of double-straned pairings, bulges (i.e. unpaired sequences within stems), terminal loops, internal loops, multiloops and so on. Since the stability of the stem is determined by its length and the number of mismatches or bulges it contains, we encode the double-stranded pairings as “P” to emphasize the steadiness of the stem structure. If there are unpaired sequences within the stem, the contents of nucleotides are preserved to accentuate the “differences” between the compared structures. Moreover, the nucleotides appeared in internal loop substructure, we encode them by label “L”. For searching IRE patterns in UTRs, the proposed algorithm is designed to work with NC-IUB code to symbolize the loop pattern, CAGWGH. We demonstrated the NC-IUB encoded IRE graph examples in Fig. 1(d).

E. Transform the Simple Graphs into Line Graphs

Before detecting the maximum common subgraph between two graphs, the proposed algorithm transforms them into line graphs. The line graph of a graph *G*, written *L*(*G*), is the graph whose vertices are the edges of *G*, with *ef* ∈*E*(*L*(*G*)) when *e* = *uv* and *f* = *vw* in *G*.

F. Construct the Compatibility Graph

The second part of this proposed algorithm is to detect the maximum common subgraph between two compared graphs. The maximum common subgraph problem can be reduced to determin the maximum clique in the compatibility graph [10]. The compatibility graph of two labeled graphs G_1 and G_2 is defined on the vertex set $V(G_1) \times V(G_2)$ with two vertices $u_i \in V(G_1)$ and $u_j \in V(G_2)$ having respective adjacent vertices $v_i \in V(G_1)$ and $v_j \in V(G_2)$, whenever

$$\begin{array}{l} (u_i, v_i) \in E(G_1) \; and \; (u_j, v_j) \in E(G_2) \\ or \\ (u_i, v_i) \notin E(G_1) \; and \; (u_j, v_j) \notin E(G_2) \end{array} \quad (14)$$

However, an increasing number of vertices in compatibility graph will dramatically increase the number of edges that are detected in the maximum clique later. The compatibility graphs are sparse, most vertices are not adjacent to the other vertices in the original graph, nevertheless, the condition of $(u_i, v_i) \notin E(G_1)$ *and* $(u_j, v_j) \notin E(G_2)$ in the compatibility graph will dominate, and the compatibility graph will be dense as well as large. As a result, Raymond *et al.* [11] has extended the definition of the compatibility graph to apply the compatibility graph to the problem of the MCS (Maximum Common Subgraph) of two chemical graphs. Since the major purpose of this proposed algorithm is to detect the MCS of two RNA secondary structures, we also have to extend the definition of the compatibility graph. Note that each vertex corresponds to a labelled nucleotide type in the original RNA secondary structure graphs, denoted as $Content\,(v_j^1)$, where *Content* indicates the type of nucleotide. As Eq. (6) shown, each vertex in two compared graphs is considered when constructing the compatibility graph. The symbol w_{ij} indicates a vertex in a compatibility graph while the content of the vertex u_i in $L(G_1)$ is equal to that of the vertex v_i in $L(G_2)$. As a result, the connection between two vertices in compatibility graph exists if both the Eqs. (14) and (15) are true.

$$\sigma_C(v_j^1, v_j^2) = 1 \; and \; \sigma_C(u_j^1, u_j^2) = 1 \quad (15)$$

When the compatibility graph is constructed, the next step is to find the maximum clique from the compatibility graph.

G. Description of Proposed Maximum Clique Detection Procedures

A clique is defined as a set of vertices in which every vertex is connected to every other vertex by an edge. The definition of the clique problem is that, given a graph *G* and an integer *k*, determine if *G* has a clique *C* of *k* vertices. The maximum clique problem is also known to be NP-complete. Our goal is to find a maximum clique and the size of the maximum clique. Once a clique is found, we only need to enumerate cliques better than the current best clique. The most well known and commonly used implicit enumerative method for the maximum clique problem is the branch and bound method. The key issues in a branch and bound algorithm for the maximum clique problem are: how to find the proper upper bound on the size of maximum clique and how to branch, which means how to break a problem into several small sub-problems [12]. As described in the previous section, we have a compatibility graph (G_1 ◊ G_2), denoted as G'_{12}, next we sort all vertices of G'_{12} according to the degrees by increasing order, say $v_1, v_2, \ldots, v_n$, where v_1 is a vertex with the smallest degree of the graph G'_{12}. The clique detection algorithm that we use and modify is developed by Carraghan *et al.* [13], which serves as a benchmark algorithm for clique finding. Initially, this algorithm finds the largest clique C_1 that contains the vertex v_1. Next it finds clique C_2, which is the largest clique in $G'_{12} - \{v_1\}$ and contains v_2 and so on. For the purpose of reducing the search space, this clique detection algorithm applies heuristics and prune techniques. The crucial factor in the algorithm is the notion of depth. Considering the vertex v_1, in depth 2, we select all vertices that are adjacent to v_1. In depth 3, we then select the vertices that are shown in depth 2 and are adjacent to the first vertex listed in depth 2. Let v_{di} be the vertex that is currently expanding at depth d and step i, and if $d + (m – i) \geq$ the size of current largest clique, then continue searching, as the possible largest clique would be larger or equal to the size of the current largest clique; otherwise, stop searching. If the procedure is at depth 1 and this inequality cannot hold then the procedure stops. However, a common feature of the sequential heuristics is that they all find only one maximal clique. Once a maximum clique is found, the search stops, however, there may be more than one maximum clique. To solve this problem, we preserve a set to keep all possible maximum cliques while performing clique finding, named the candidate clique set; this is the set of cliques that have been already detected as the largest clique so far. Initially, this set is empty until the first clique is found, then the algorithm continues to check whether any clique exists, if the result is yes and the size of the newly found clique is larger than that in candidate clique set, then all cliques in the candidate clique set are removed and the larger clique is stored. If the newly found clique is the same size as the candidate clique set, then store this clique in the candidate clique set. We illustrate this with an example paeudo-code. The main purpose of the candidate clique set is to preserve all possible maximum cliques, and then to manually choose one that conforms to the biological meanings. We summarize the pseudo-code of whole algorithm in appendix section.

III. Results

A. Performance on Precursor microRNA Secondary Structures

We perform first experiment by using Let-7 precursor microRNA [14][15] secondary structures. In Fig. 2, we demostrate the maximum common subgraphs detected by our proposed algorithm for the precursor microRNAs compared, and the blue circles shown in Figs. 2(a), 2(b), and 2(c) indicate the maximum common secondary structures between cel-let-7 and human let-7a-1, let-7b, and let-7c precursor microRNAs, respectively. The experimental results validate our expectation that if the terminal loops, internal loops or bugles have different sizes in compared secondary structures, they will not be seen in the results. As a result, with Fig. 2(a) as an example, both cel-let-7 and hsa-let-7a-1 structures have internal loops in stems, with lengths of 2nt and 4nt, respectively, and neither are shown in the result of the common substructures. Looking at Fig. 2(c), both cel-let-7 and hsa-let-7c structures have the same internal loops, 2nt in length and with the same contents of nucleotides (the nucleotide U). Hence, the proposed algorithm regards the internal loops as the same structures and extends the scope when detecting common subgraphs. This leads to an interesting phenomenon, if the internal loops have the same length but different contents of nucleotides, does the proposed algorithm detect these internal loops as the same structure? The answer is negative, as illustrated in Figs. 2(d) and 2(e), cel-let-7, hsa-let-7d and hsa-let-7e all have internal loops with length of 2nt but different contents of nucleotides, and the results demonstrate that these internal loops are different structural elements. The reason for causing this experimental result is Eq. (15), which is the extended definition of the compatibility graph. Accordingly, we execute another experiment for further confirmation. We encode all nucleotides that appeared in the internal loop structural elements with symbol “L” and encode the nucleotides appeared in stems with symbol “P”. The experimental results substantiate our doubts and are indicated by the red circles shown in Figs. 2(d) and 2(e), which indicates that if the internal loops have the same length, they can be detected as similar structural elements by the encoding procedure. Additionally, it remains ambiguous that whether the biological meanings or functions of internal loops influence the binding effects of the RNA binding proteins. Therefore, we encode the loops with symbol “L” to make this proposed algorithm more flexible to extract loops with same sizes of length. If the biologists concern the contents of nucleotides in loops, this proposed algorithm also has the ability of extract the “exact match” structural elements without encoding procedure.

We present the similarity scores between cel-let-7 and other precursor microRNAs in Fig. 2(f). We compare human let-7a-1, let-7b, let-7c, let-7d, and let-7e to *C.elegans* let-7 precursor microRNA, respectively. The similarity scores are calculated with the parameters $\alpha = 0.8$ and $\beta = 0.2$, and encoding symbol “P” in the stem substructure and internal loop encoding by symbol “L” *not* enabled are listed in the second column of Fig. 2(f). Notice that, the highest score in this column is 0.7099, which indicates the hsa-let-7c precursor is most similar to cel-let-7 precursor by measuring the labels of vertices (Eq. 3), the labels of adjacent vertices (Eqs. 4-6), and the labels of edges (Eqs. 7-11). That is, the proposed scoring method considers not only the variety of nucleotides (the lables of vertices), but also the diversity of double-stranded pairings (the labels of edges). The similarity scores listed in the last column have encoded the double-stranded pairings by label “P” to emphasize the steadiness of the double-stranded pairings, and encoded by label “L” in internal loops substructure. As a result, the similarity scores are higher than those listed in second column. The higest score listed in the third column is 0.9709, which indicates that after the encoding precedure, the more double-stranded pairings, the higher scores calculated by this proposed scoring measurement. In conclusion, we provide experimental evidence to prove the capability of the proposed algorithm in measuring similarity scores and detecting maximum common structures for precursor microRNAs, and in the next section, we use RNA secondary structures with multiloops to verify the accuracy of our proposed algorithm.

B. Performance on RNA Secondary Structures with Multiloops

The purpose of this experiment is to test how accurate the proposed algorithm is for more complex secondary structures. We select ribosomal 5S ribonucleic acid (5S rRNA) which has a length of ~120nt for detailed test because the 5S rRNA has more complicated secondary structural elements such as hairpins, internal loops, bulges, and multipl-loops. 5S rRNA is a component of the ribosomal subunit in both prokaryotes and eukaryotes. The function of the rRNA is to provide a mechanism for decoding mRNA into amino acids and to interact with the tRNAs during translation. Fig. 3 demonstrates the maximum common substrucutre of two compared 5S rRNA secondary structures. One is Methanococcus vannielii 5S rRNA (accession number X02729/3000-3137 in RFAM database) and the other one is Desulfurococcus mobilis 5S rRNA (X07545.1/505-619) [16]. We use the same encoding procedure described in the previous section to transform the nucleotide labels into “L” and “P” for those located in internal loops and double-stranded pairings, respectively. Graphically, the right stem-loop substructure of Fig. 3(a) seems to have some resemblance to that of Fig. 3(b), nevertheless, the actual secondary structure is quite diverse in the length of stem and the end loop. That is, the proposed algorithm performs precisely when comparing the different size of loops. Specifically, as illustrated by the red marks in Fig. 3, only stems with the same number of double-strand pairings and the loops with the same sizes can be detected as similar structural elements. It is remained ambiguous that whether the sizes (the length of nucleotides) or mutations (the different contents of nucleotides) of multiloops influence the biological functions effect on complex RNA secondary structures.

We also provide the experimental results of extracting maximum common structural elements between Homo sapiens and mouse U12 minor spliceosomal RNA secondary structures in Fig. 4. U12 minor spliceosomal RNA is formed from U12 snRNA, and it works with U4atac, U5 snRNA, U11 snRNA and related proteins to form a spliceosome which cleaves a class of low-abundance pre-mRNA introns [17]. Both L43846.1 (Homo sapiens) and L43843.1 (mouse) sequences are also downloaded from RFAM database. As illustrated in Fig. 4, the conserved secondary structural element is marked by red circles, which is identified by our proposed algorithm as well. It is interesting that after coding the vertices by label P and L (for nucleotides in loops), the proposed algorithm regards the left three stem-loops appeared in Fig. 4(a) and 4(b) as common subgraph. This proposed algorithm provides a detailed observation of complex RNA secondary structures by extracting maximum common substructures on the basis of graph theoretical approach, which is distinct from tree-liked model described in previous section.

C. The Experimental Results of Working with NC-IUB Code

We use proposed algorithm to search for RNA hairpins in UTR regions of human mRNA sequences. This experiment is designed with a subject sequence containing an iron response element (IRE). The IRE is a hairpin (stem loop) structure containing about 30 nucleotides. The IRE is a highly conserved RNA hairpin structure, and it is the binding site of iron regulatory protein (IRP). IRP binding to IRE is regulated by cellular iron. When cells are derived of iron, IRP binds IRE. If IRE is located at 5'UTRs, IRP binding will inhibit translation initiation; otherwise, if IRE is at 3'UTRs, IRP binding will stabilize mRNA and prevent it from degradation [18]. Using a subsequence with 50 nucleotides in the 3’UTR of transferring receptor gene (NM_003234), which contains the IRE hairpin, we search for “similar” hairpin structures in the human UTR structure database, as described in method section. We query database for gene sets that contain one of the Gene Ontology (GO) terms “iron ion transporter activity”, “iron ion binding”, “iron-responsive element binding”, and “iron-sulfur cluster binding”, and there are 25,722 genes found. The purpose of this experimental design is to understand if one gene has functional annotations related to “iron ion”, does it have similar IRE structure patterns in the UTR regions of mRNAs. It is known that the gene TFRC (NM_003234) has been found to have nine stem-loop structures in the 3’UTR part, and at least five of them are confirmed to be related to the iron response mechanism, which are listed as follows: CAGUGU, CAGUGC, CAGUAU (CAGUGU), CAGUGA, and CAGUGU. Since we know that the IRE is highly conserved in the loop sequence, we can form the loop pattern with the NC-IUB code, represented as “CAGWGH”. As for the double-stranded pairings (the stem), mutation in this region may not be injurious to the function of an RNA if the mutated nucleotides still preserve the same secondary structure. Therefore, the nucleotides in the double-stranded helix part are encoded by NC-IUB code “N” while searching the structure database for similar IRE patterns. Following the folding procedures which are discussed in method section, we fold the UTR sequences to form the experimental data set. We calculate the similarity scores between the TFRC IRE pattern and all hairpin structures for the 5’UTR and 3’UTR parts of the 25,722 genes. Next we filter those which have a similarity score below 0.6 with the parameters $\alpha = 0.85$ and $\beta = 0.15$. Collectively, this proposed algorithm finds 29 hits (excluding the subject structure, NM_003234), among which six are known true positives. The list of hit structures is shown in Table I.

IV. Discussion

The construction of a purely structural based approach for comparing RNA secondary structures is one of the most important problems in computational biology. The experiment of the IRE hairpin searching can be regard as a pattern-based detection method, that is, our proposed algorithm can work with NC-IUB code to search for particular hairpin patterns. Since this algorithm considers the labelling of vertices and edges while constructing the compatibility graph, if the length of the end loop, internal loop or double-stranded pairings is different between two compared structures, our proposed algorithm regards them as different structure and excludes them, and this is why our proposed algorithm is better at finding common shapes. As for the time complexity of the proposed algorithm, the compatibility graph can be recognized in time $O(m\times n)$ for graph $L(G_1)$ with m vertices and $L(G_2)$ with n vertices. The operation is essentially commutative, as the graphs $G_1 \lozenge G_2$ and $G_2 \lozenge G_1$ are isomorphic. After constructing the compatibility graph, the proposed clique detection procedures are performed. The first step in performing clique detection is sorting the vertices by degrees in increasing order, for which the Quicksort algorithm [19] is used. The average time complexity for sorting n items is $O(n\log n)$, however, in the worst case, it takes $O(n^2)$ in time complexity. After sorting, the procedures of clique detecion are started. It is known that the maximum clique problem is one of the first problems shown to be NP-complete, and is supposed to be false, unless $P = NP$, an exact algorithm is ensured to return a solution only in a time that increases exponentially with the number of vertices in the graph. As a result, the interest in this problem has moved toward approximation. Garey and Johnson [20] prove that if the maximum clique problem permits an algorithm that is a polynomial-time approximation, then it is able to be approximated within any arbitrarily factor. If this is one of the solutions to the complete problems, then the optimum solution can be approximated to arbitrarily small constant factors. Suppose we know the maximum clique has size ≥ γ, the clique detection algorithm that we use may use γ as pruning and stopping criterion which is $d+(m-i)<\gamma$. If γ is close to the size of the actual maximum clique, the computational time can be reduced for graphs with high density. Turan [21] have proved that a graph with n vertices and m edges contains a clique of size $\geq n^2/(n^2-2m)$. As a result, with the density D of the graph, we may have $\gamma = n/(1-D)\times n + D$. However, the value of γ is helpful only if it is very close to the actual maximum clique size, and may be solved in polynomial time. When this proposed algorithm performs the converted clique finding procedures, it does not stop when one maximum clique has been found. In the contrary, it continues to search for more maximum cliques. We design a preserved set to keep all possible maximum cliques, named the candidate clique set, which is a set of cliques that have already detected the largest clique so far. The main purpose of candidate clique set is to preserve all possible maximum cliques, and then we choose one that confirms biological meanings manually. Experimentally, this proposed algorithm has proved its performance in extracting the maximum common subgraph in RNA secondary structures. This work presented here is intended to provide a method to directly perform the structural comparison of RNA secondary structures, and its capability to identify common substructure can potentially be used to predict the functions of RNA structural elements.

V. Conclusion

A graph-based model for comparing purely RNA secondary structures has been proposed based on finding the maximum common subgraph between the graphs being compared without transforming RNA secondary structures into tree graphs. It is a new idea of comparing RNA secondary structures without representing by tree-liked models but using simple graphs instead. Moreover, this proposed algorithm is divided two major parts, the calculation of similarity score and the detection of the maximum common secondary structures between two compared RNA secondary structures. In the calculation of similarity scores part, this algorithm considers not only the cost distance according to the numbers of vertices and edges displayed in G_1 and G_2, but also takes account of the labels of vertices and edges. In the detection of maximum common secondary structure part, the proposed algorithm represents secondary structures by simple graphs, converting them into line graphs, constructing a compatibility graph of two line graphs, and detecting the maximum clique from the compatibility graph to find the maximum common subgraph. We have proved the performance of the proposed algorithm by providing the experimental results of precursor microRNAs let-7, 5s rRNA, U12 minor spliceosomal RNA and the detection of IRE patterns in UTRs. While the proposed algorithm has been designed for use in detection of common RNA secondary structures, it is directly applicable to other graph-based similarity applications, such as protein secondary structures.

It may be possible to significantly improve the performance of the algorithm by incorporating other clique detection techniques, that is, future work on the algorithm may focus on the calculation of similarity for protein structures, as well as its application to protein structure problems such as searching and the prediction of biological functions.

Name : Transferrin IRE (length 29)
RNA sequences : GAUUAUCGGGAGCAGUGUCUUCCAUAAUC
Folding structure: ((((((. (((((. . . . . .)))))))))))
minimum free energy = -8.40 kcal/mol

Name : mt-ACONITASE IRE(length 29)
RNA sequences : CCUCAUCUUUGUCAGUGUACAAAAUGGCG
Folding structure : . . ((((.(((((. . . . . .))))))))) . .
minimum free energy = -2.60 kcal/mol

(a)

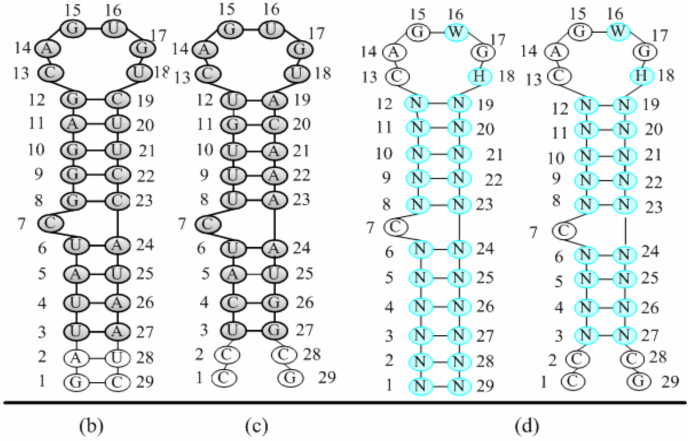
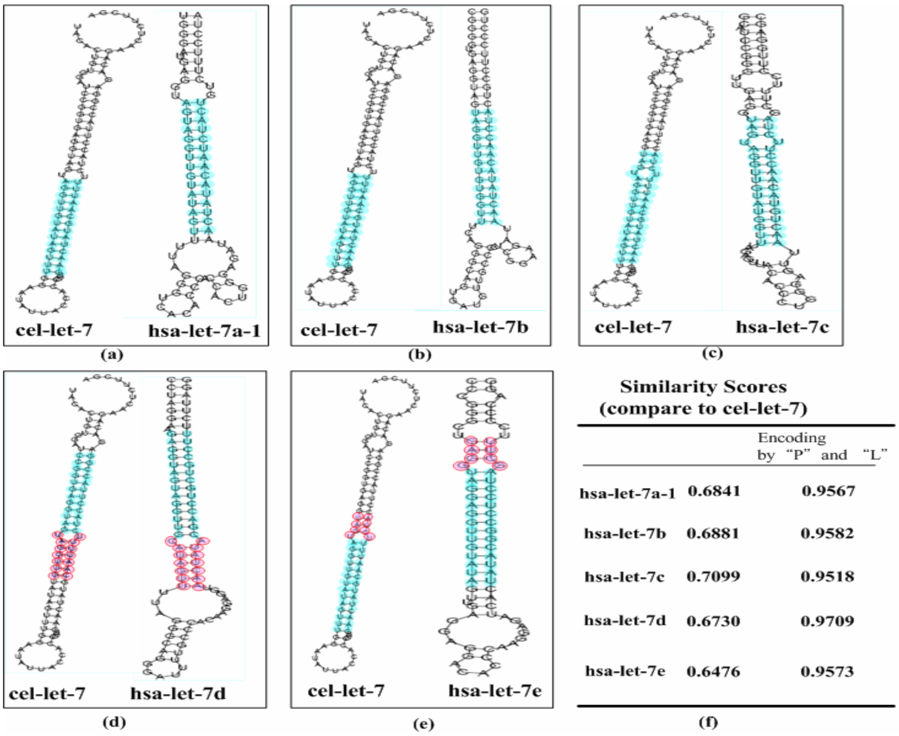


Fig. 1 Examples of IRE secondary structures (a) Two examples of IRE patterns. Both are 29nt and folded by the RNAFold program. (b) and (c) illustrates the simple graphs that are transformed from RNA secondary structures. The gray vertices and darker edges indicate the maximum structure detected by our proposed algorithm. (d) Demonstrates the results of encoding by NC-IUB code in double-stranded pairings and loop sequences, which is marked as blue circles



Similarity Scores
(compare to cel-let-7)

		Encoding by "P" and "L"
hsa-let-7a-1	0.6841	0.9567
hsa-let-7b	0.6881	0.9582
hsa-let-7c	0.7099	0.9518
hsa-let-7d	0.6730	0.9709
hsa-let-7e	0.6476	0.9573

Fig. 2 Experimental results of precursor MicroRNAs secondary structures. (a)(b)(c)(d)(e) show the maximum common subgraph (indicated by the blue and red color circles) between cel-let-7 and hsa-let-7a-1, hsa-let-7b, hsa-let-7c, hsa-let-7d, and has-let-7e, respectively. The red circles represent the results of encoding an internal loop while detecting the maximum common subgraph. After encoding the nucleotides in internal loop structures, the proposed algorithm is able to detect a larger maximum common subgraph, which is more suitable for the biological meanings. (f) Demonstrates the calculated similarity scores between cel-let-7 and hsa-let-7a-1, hsa-let-7b, hsa-let-7c, hsa-let-7d, and has-let-7e, respectively. In the last column of the table, the scores is calculated after encoding nucleotides by label "P" (for nucleotides appeared in double-stranded pairings) and "L" (for nucleotides appeared in loops)

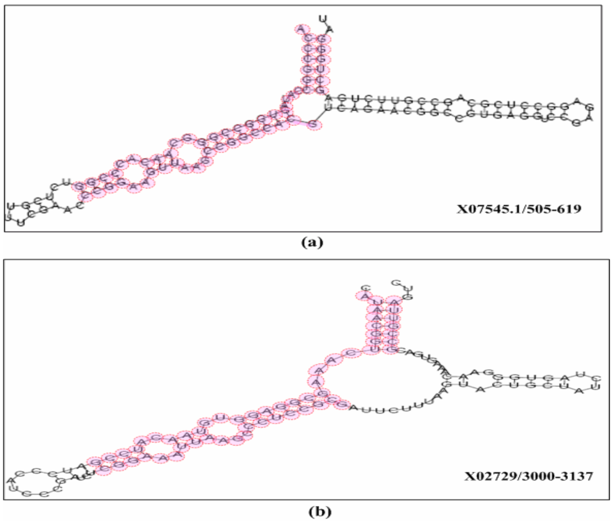


Fig. 3 Experimental results of 5S rRNA secondary structures. (a)(b) Demonstrating the 2D secondary structures for X07545 (the up diagram), and the red circles indicate the common sub-structures the same as X02729(the bottom digram) with encoding N in double-stranded pairings and L in an internal loop. Since there are two internal loops with the same size between X07545 and X02729, the proposed algorithm regards them as the same structures. As for the right stem-loop sub-structure of X07545 and X02729, they have different lengths in double-stranded pairings and different sizes of internal loop. Therefore, our proposed algorithm eliminates the right sub-structure from common structures

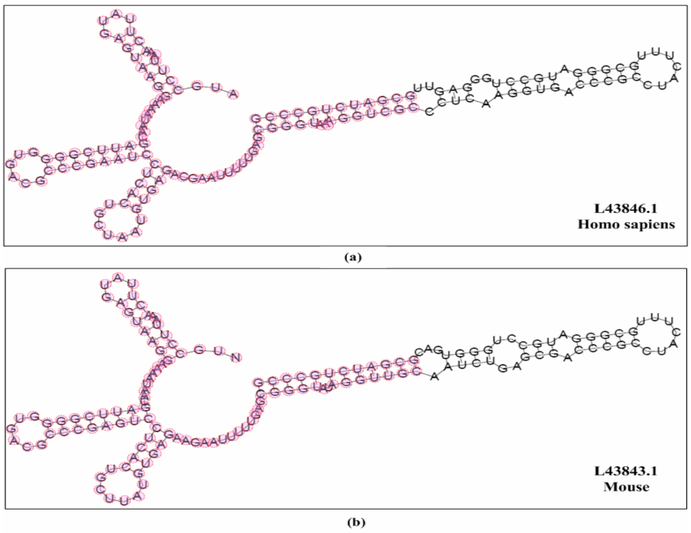


Fig. 4 Experimental results of U12 minor spliceosomal RNA secondary structures. (a) The L43846.1 (Homo sapiens) RNA secondary structure. (b) The L43843.1 (mouse) RNA secondary structure. As illustrated in (a) and (b), the maximum common secondary structural element is marked by red circles, which is identified by our proposed algorithm as well

TABLE I
THE EXPERIMENTAL RESULTS OF IRE STRUCTURES SEARCHING

Accession num	Official Name	UTR	position	Free energy	Similarity score	True Positive
NM_003234	uransferrin recepuor (p90, CD71)	3’	1349	-10.8	1.0	√
NM_173649	hypouheuical prouein FLJ40172	3’	812	-18.5	0.89	
NM_000456	sulfite oxidase	3’	413	-11.6	0.84	
NM_003234	uransferrin recepuor (p90, CD71)	3’	896	-8.2	0.83	√
NM_000617	SLC11A2 solute carrier family 11 (proton-coupled divalent metal ion transporters), member 2	3’	39	-9.9	0.82	
NM_014585	solute carrier family 40 (iron-regulated transporter), member 1	5’	213	-11	0.81	√
NM_005536	inosiuol(myo)-1(or 4)-monophosphauase 1	3’	337	-6.9	0.81	
NM_024076	pouassium channel ueuramerisauion domain conuaining 15	3’	551	-13.8	0.81	
NM_003234	uransferrin recepuor (p90, CD71)	3’	946	-8.6	0.78	√
NM_014585	soluue carrier family 40 (iron-regulaued uransporuer), member 1	5’	213	-16.6	0.78	
NM_000146	ferritin, light polypeptide	5’	38	-9.9	0.76	√
NM_002481	prouein phosphauase 1, regulauory (inhibiuor) subuniu 12B	3’	5712	-8.5	0.75	
NM_004921	chloride channel, calcium acuivaued, family member 3	3’	954	-8.2	0.75	
NM_014930	zinc finger prouein 510	3’	124	-8.3	0.74	
NM_002081	glypican 1	3’	1558	-19.3	0.73	
NM_004109	ferredoxin 1	3’	363	-8.9	0.73	
NM_003449	uriparuiue mouif-conuaining 26	3’	337	-9.8	0.71	
NM_003234	uransferrin recepuor (p90, CD71)	3’	1414	-8.1	0.70	√
NM_018234	SUEAP family member 3	3’	1734	-10.5	0.70	
NM_018992	pouassium channel ueuramerisauion domain conuaining 5	3’	1433	-16.3	0.70	
NM_032484	GH3 domain conuaining	3’	328	-15.9	0.67	
NM_015219	exocysu complex componenu 7	3’	900	-12	0.66	
NM_017637	basonuclin 2	3’	7119	-5.7	0.66	
NM_001013839	exocysu complex componenu 7	3’	900	-12	0.66	
NM_015219	exocysu complex componenu 7	3’	411	-16.4	0.65	
NM_003234	uransferrin recepuor (p90, CD71)	3’	1461	-13.3	0.64	√
NM_001009909	leucine zipper prouein 2	3’	1224	-8.9	0.64	
NM_001013839	exocysu complex componenu 7	3’	411	-16.4	0.64	
NR_002787	hypouheuical prouein LOC154449	3’	383	-6.1	0.63	
NM_198316	uensin like C1 domain conuaining phosphauase (uensin 2)	3’	406	-15.3	0.61	

APPENDIX

The method we introduce is divided in two major parts: The main program and the function named *clique_finding*.

```
Main Program
Input: primary RNA sequences
Output: the similarity score and the maximum common substructure
BEGIN
Step 1: Form the secondary structures by the RNAFold program which outputs the combinations of parentheses and dots. The proposed algorithm
        transforms the combinations of parentheses and dots into a simple graph.
Step 2: Calculate the similarity score of two compared structures by using Eqs. (1) ~ (13). Record the similarity score.
Step 3: Encode the graphs by label $P$ and $L$, which is described in previous section. Transform the simple graphs into line graphs. For example,
        each vertex in the line graph $L(G)$ is labelled with its respective edge and vertex endpoint labels in the graph $G$. The edges in a labelled
        line graph are also labelled.
Step 4: Construct the compatibility graph with Eqs. (14) and (15).
Step 5: Call function clique_finding to detect the maximum clique in the compatibility graph.
END

Function clique_finding
Initialize the parameters:
   Set parameter $v_{di}$ (the vertex that is currently expanding at depth $d$ and step $i$) as null.
  Clear NodeDegreeSort (a parameter to save the degree of each vertex)
  Clear parameter CSS (Candidate Clique Set)
  Clear parameter NMC (Number of Maximum Clique)
  Set $m = 0$
Begin:
$d = 1$
Increasing sort of all vertices in $G$ according to the degree of each vertex
Save the results of sorting in NodeDegreeSort
FOR $j = 1$ to length of NodeDegreeSort
   DO
      Depth $d$:    assign $j^{th}$ element of NodeDegreeSort to $v_{di}$
                    now expand $v_{di}$
      Depth $d$++: consider all vertices that adjacent to $v_{di}$ in increasing order
                                $m$ = the number of vertices that adjacent to $v_{di}$
                         such as $v_{d1}, v_{d2}, \ldots, v_{dm}$ (where $d$ is the current depth)
                           each vertex performs the expanding procedures
                           call function check_stop
                         IF (return value of check_stop is true) continue
                         ELSE break
           Depth $d$++: consider vertices appeared in last depth that are adjacent to the first vertex in last depth
                         $m$ = the number of vertices appeared in last depth and are adjacent to the first vertex in last depth
                         call function check_stop
                         IF (return value of check_stop is true) continue
                         ELSE break
     UNTIL   can not expand from last depth.
      Temp_NMC = $d$ (the depth)
      Call function check_reset
     Call function check_stop
     Reset d = 1
     IF (return value of check_stop is true) continue
     ELSE break loop
END OF FOR loop
End of function clique_finding

Function check_stop
   IF $(d+(m-i) \geq \text{NMC})$ return True
   ELSE return False
End of Function check_stop

Function check_reset
   IF Temp_NMC > current NMC
      reset new value of NMC by Temp_NMC
      clear CCS
      save new max clique in CCS
      return to main program
   IF Temp_NMC = current NMC
      Save newly found clique in CCS
      return to main program
End of Function check_reset
```

REFERENCES

[1] M. Hochsmann, B. Voss, and R. Giegerich, "Pure multiple RNA secondary structure alignments: a progressive profile approach," *IEEE Trans. on Computational Biology and Bioinformatics,* vol. 1, pp. 53-62, 2004.

[2] J. Liu, J. T. L. Wang, J. Hu, and B. Tian, "A method for aligning RNA secondary structures and its application to RNA motif detection," *BMC Bioinformatics*, vol. 6, pp. 89-109, 2005.

[3] J. Allali and M. F. Sagot, "A new distance for high level RNA secondary structure comparison," *IEEE Trans. on Computational Biology and Bioinformatics*, vol. 2, pp. 1-11, 2005.

[4] G. D. Collins, S. Le and K. Zhang, "A new algorithm for computing similarity between RNA structures," *Information Sciences*, vol. 139, pp. 59-77, 2001.

[5] S. Dulucq and L. Tichit, "RNA secondary structure comparison: exact analysis of the Zhang-Sasha tree edit-algorithm," *Theoretical Computer Science*, vol. 306, pp. 471-484, 2003.

[6] J. T. L. Wang, B.A. Shapiro, D. Shasha, K. Zhang and K.M. Currey, "An algorithm for finding the largest approximately common substructures of two trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 889-895, 1998.

[7] I. L. Hofacker, "Vienna RNA secondary structure server," *Nucleic Acids Research*, vol. 31, pp. 3429-3431, 2003.

[8] T.J. Macke, D. J.Ecker, R.R. Gutell, D. Gautheret, D. A. Case and R. Sampath, "RNAMotif, an RNA secondary structure definition and search algorithm," *Nucleic Acids Research*, vol. 29, pp. 4724-4735, 2001.

[9] K. D. Pruitt, T. Tatusova and R. D. Maglott, "NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins," *Nucleic Acids Research,* vol. 35(Database issue), pp. D61-65, 2007.

[10] G. Levi, "A note on the derivation of maximal common subgraphs of two directed or undirected graphs," *Calcolo*, vol. 9, pp. 347-352, 1973.

[11] J. W. Raymond and P. Willett, "Maximum common subgraph isomorphism algorithms for the matching of chemical structures," *Journal of Computer-aided Molecular Design*, vol. 16, pp. 521-533, 2002.

[12] P. Pardalos and J. Xue, "The maximum clique problem," *J. Global Optimiz*, vol. 4, pp. 301-328, 1994.

[13] R. Carraghan and P. M. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Research Letters*, vol. 9, pp.375-382, 1990.

[14] N. C. Lau, L. P. Lim, E.G. Weinstein and D. P. Bartel, "An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans," *Science*, vol. 294, pp. 858-862, 2001.

[15] L. P. Lim, N.C. Lau, E.G. Weinstein, A. Abdelhakim, S. Yekta, M. W. Rhoades, C.B. Burge, D.P. Bartel, "The microRNAs of Caenorhabditis elegans," *Genes & Development*, vol. 17, pp. 991-1008, 2003.

[16] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna and S. R. Eddy, "Rfam: and RNA family database," *Nucleic Acids Research*, vol. 31, pp. 439-441, 2003.

[17] L. R. Otake, P. Scamborova, C. Hashimoto, J. A. Steltz, "The divergent U12-type spliceosome is required for pre-mRNA splicing and is essential for development in Drosophila," *Molecular Cell*, vol. 9, pp. 439-446, 2002.

[18] N. C. Andrews, "Disorders of iron metabolism," *New England Journal of Medicine*, vol.341, pp. 1986-1995, 1999.

[19] C. A. R. Hoare, "Quicksort," *Computer Journal,* vol. 5, pp. 10-15, 1962.

[20] M. R. Garey and D. Johnson, "The complexity of near-optimal graph coloring," *Journal of the Association for Computing Machinery*, vol. 23, pp. 43-49, 1976.

[21] P. Turan, "On the theory of graphs," *Colloq. Math*, vol. 3, pp. 19-30, 1954.