BIOINFORMATION
Discovery at the interface of physical and biological sciences

open access www.bioinformation.net Prediction model

Volume 7(2)

Combinatorial permutation based algorithm for representation of closed RNA secondary structures

Athanasios T Alexiou*, Maria M Psiha, Panayiotis M Vlamos

Department of Informatics, Ionian University, Plateia Tsirigoti 7, 49100 Corfu, Greece; Athanasios T Alexiou - Email: alexiou@ionio.gr, Phone: +30 6944 551797, Fax: +30 210 9533296; *Corresponding author

Received August 22, 2011; Accepted August 30, 2011; Published September 06, 2011

Abstract:
A permutation-based algorithm is introduced for the representation of closed RNA secondary structures. It is an efficient 'loopless' algorithm, which generates the permutations on base-pairs of '*k*-noncrossing' setting partitions. The proposed algorithm reduces the computational complexity of known similar techniques in *O(n)*, using minimal change ordering and transposing of not adjacent elements.

Keywords: Closed RNA secondary structures, *k*-noncrossing partitions, permutation-based algorithm

Background:
The bimolecular structure prediction problem has been examined for years, based on the fact that a function of a bimolecule is largely dictated by its structure. The ultimate goal of structure prediction is to obtain the three dimensional structure of bimolecules through computation. The key concept for solving the above mentioned problem is the appropriate representation of the biological structures. The problems that concern representations of bimolecular structures are either characterized as NP-complete or with high complexity. A characteristic common to these problems of molecular biology consists in the satisfaction of a set of constraints coming from different sources of biological knowledge. Hence, we focus on the representation and visualization of closed RNA secondary structure without pseudoknots, which can reasonably be viewed as a first step towards three dimensional prediction modeling. Generally, there are six kinds of representations for closed RNA secondary structures: Full representation, Tree representation, Circle representation, Arc annotated, Mountain representation and Bracket representation. The major areas of computational study in RNA secondary structure prediction include dynamic programming algorithms **[1],** stochastic algorithms such as Bioambients calculus **[2],** comparative methods **[3],** simulated annealing **[4],** and most recently evolutionary algorithms which attempt to mimic a natural folding pathway by using a populations based approach **[5].** Nowadays, an increasing number of researchers have released novel RNA structure analysis and prediction algorithms for comparative approaches to structure prediction. Their approaches are based on the fact that closed RNA structures can be viewed as mathematical objects obtained by abstracting topologically non-relevant properties of planar folding of single-stranded nucleic acids. These algorithms require significant computational resources and thus are impractical for sequences of even modest length.

From the biological view, the RNA's structure is dominated by base-pairing interactions, most of which are Watson-Crick pairs between complementary bases. The base-paired structure of RNA is called, its secondary structure. Due to the fact that Watson-Crick pairs are such a stereotyped and relatively simple interaction, accurate RNA secondary structure prediction appears to be an achievable goal. RNA secondary structures **(Figure 1)** folding cooperatively allow the creations of pseudoknot free secondary structures, where no base pairs overlap, that is there are no pair of bases (i, j) and (i′, j′) with $i < i' < j' < j$. In literature **[6]** except hairpin and interior loops we can also find definitions for bans, multiloops, external loops,

pseudo knot loops, interior-pseudo knotted loops and multi-pseudo knotted loops.

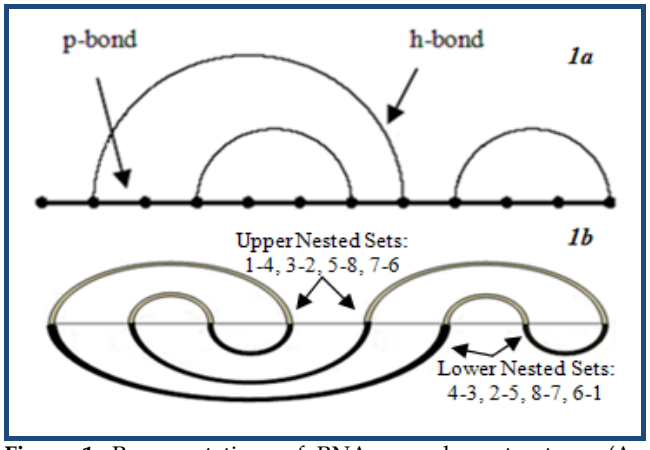

Figure 1: Representations of RNA secondary structures (An RNA molecule can be viewed as an ordered sequence of n bases and secondary structures can be generally defined as a set of pairs i - j, $1 \le i \le j \le n$, indexed starting at 1 from the so-called 5`-end and with each index in, at most, one pair.) **(a)** A secondary structure can be represented as an arc diagram, in which base indices are shown as vertices on a straight line, ordered form the 5`-end and arcs (always above the straight line) indicate base pairs, and all chemical bonds of its backbone are ignored. **(b)** Matching Nested Sets as an example of permutation [1-4-3-2-5-8-7-6] in M_4.

Methodology:

In this case consideration will be given to the surveys of Trotter **[7]** & Johnson **[8]** for the generation of specific permutations by transposing pairs of elements, using a recursive procedure.

K-noncrossing closed RNA structures:

Closed RNA secondary structure is represented as k-noncrossing set of partitions, which corresponds to the base-pairs and no base-pairs respectively. A (set) partition of [2n] is a collection of disjoint subsets on [2n], representing a 2n union **(Figure 1a).** Each element of a partition is called a block. A (complete) matching on [2n] = {1, 2, 2n} can be represented by listing its $2n$ blocks, as $\{(i_1, j_1), (i_2, j_2), \ldots, (i_{2n}, j_{2n})\}$, where $i_r < j_r$ for $1 \le r \le n$. Two blocks (also called arcs) (i, j) and (i′, j′) form a crossing if $i < i' < j < j'$, and a nesting if $i < i' < j' < j$. It is well-known that the number of matching's on [2n] with no crossings (or with no nestings) is given by the n-th Catalan number. Let π_{2n} denote the set of partitions of [2n] and a diagram $\pi \varepsilon \pi_{2n}$. A k-distant (k is a nonnegative integer) crossing of π is a pair of edges (i, j) and (i′, j′) of π satisfying $i < i' < j < j'$ and $j < i' \ge k$. A k-distant nesting of π is a set of two edges (i, j) and (i′, j′) of π satisfying $i < i' \le j' < j$ and $j < i' \ge k$. A partition or matching π is k-distant noncrossing if π has no k-distant crossing and k-distant non-nesting if π has no k-distant nesting.

Generating Permutations:

Our case study includes all the numbers that begin with 1 and have unique alternate even-odd digits. The problem mainly concerns the quick development of a special set of permutations G_{2n}, rather than the common n! permutations of the first n components. These alternative permutations can be defined in the form as shown in **supplementary material.**

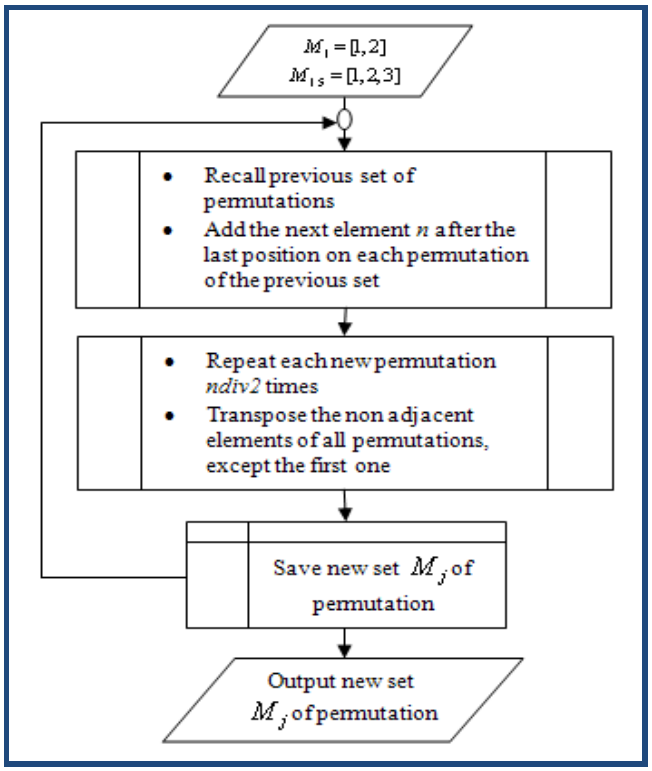

Figure 2: Diagrammatical representation of the proposed algorithm

Discussion:

RNA pseudoknot structures can be categorized in terms of the maximal size of sets of mutually crossing bonds. A k-noncrossing RNA structure has at most k-1 mutually crossing bonds and a minimum bond-length of 2, i.e., for any i, the nucleotides i and i+1 cannot form a bond. According to this formulation, a k-noncrossing RNA structure can be represented as a digraph in which all vertices have degree, that does not contain a k-set of mutually intersecting arcs and 1-arcs, i.e. arcs of the form (i, i+1), respectively **[9]**. Furthermore, RNA secondary structure is often assumed to be sufficient for being able to predict the RNA function. This assumption can be justified by observations of well conserved secondary structures and the fact that secondary structures fold fast, while tertiary interactions need much more time to form **[10].** The fact that it is possible to predict secondary structures using nearest-neighbour parameters **[11]** also suggests that secondary structure contributes much more to the stability of the RNA structure, than the tertiary interactions.

Moreover, an imperative algorithm for generating combinatorial objects is called loopless, if for every set of n elements the number of steps needed to generate the first object is less than O(n), the decision whether an object is the last is obtained within O(1) steps and every transition between successive objects requires at most O(1) steps. Generally, an algorithm is loopless if the objects are represented in a simple form and can be read directly without requiring any additional steps.

ISSN 0973-2063 (online) 0973-8894 (print)
Bioinformation 7(2): 91-95 (2011)

The proposed model-algorithm **(Figure 2)** includes the following procedure: Given an integer array of certain length (L), the algorithm generates the permutation of digits {1, 2,…, 2n} in the integer array M[i, j], i, j = 1, 2,…, 2n. The number of the iterations T is calculated proportionally with the total number of elements required for the permutations of number n. (e.g. for 3 one element, for 4→2, for 5→4 etc). A permutation is created by swapping the newly added digit n with an existing digit in the array. If n is odd, the permutation should occur only if the corresponding swapped number is odd and vice versa. Thus, only digits at positions n-3, n-5... should be considered. Note that the first digit (1) of the array is not swappable. For the n+1 element that is added after the last position on each of the previous permutations every permutation of previous mark is recalled. Since the recursive detection of the transposing, through the minimal change of permutations, can be performed at the same time, the running time of the algorithm will be proportional to the size of the computation tree (the number of recursive calls). Furthermore, in this tree, each node has exactly T-1 children and each leaf corresponds to a unique permutation.

Conclusion:

From the set of canonical pairs, it is clear that a given RNA sequence has many potential structures. In fact, the number of possible structures grows exponentially with the length of the RNA sequence. The challenge is to identify whether structure plays a functional role for a given RNA sequence and, if yes, to predict this functional RNA structure. In medical applications, accurate structural knowledge will be the starting point to create new lead compounds which would eventually be applied into more effective drugs. Therefore, the accurate prediction of RNA structure could simultaneously provide clues for curing an assortment of diseases, especially those that are based on RNA viruses. Since the conception of permutation to the individual representation of RNA secondary structure in genetic algorithms has been introduced, the problem can be essentially represented as a neural network in future work, which can be optimized through genetic algorithms techniques.

Reference:

[1] Zuker M. *Science* 1989 **244**: 48 [PMID: 2468181]
[2] Regeva A *et al. Theoretical Computer Science* 2004 **325**: 141
[3] Mathews DH. *Turner DH Biochemistry* 2002 **41**: 869 [PMID: 11790109]
[4] Schmitz M & Steger G. *J Mol Biol.* 1996 **255**: 254 [PMID: 8568872]
[5] Wiese KC & Hendriks A. *Bioinformatics* 2006 **22**: 934 [PMID: 16473869]
[6] Rastegari B. *Condon A Springer WABI.* 2005 **3692**: 341
[7] Trotter HF. *ACM* 1962 **5**: 434
[8] Johnson SM. *Mathematical Compinatorics* 1963 **17**: 282
[9] Jin EY *et al. Bull Math Biol.* 2007 **70**: 45 [PMID: 17896159]
[10] Onoa B & Tinoco I Jr. *Curr Opin Struct Biol.* 2004 **14**: 374 [PMID: 15193319]
[11] Mathews DH & Turner DH. *J Mol Biol.* 2002 **317**: 191 [PMID: 11902836]

Edited by P Kangueane
Citation: Alexiou *et al.* Bioinformation 7(2): 91-95 (2011)

BIOINFORMATION

open access

Supplementary material:

An alternative permutation can be defined in the form:

$M_{2n} = 1\pi_2\pi_3...\pi_n$

Where π_i exists once in M_{2n} and

$$\pi_i = \begin{cases} even, \text{if } i \text{ even} \\ odd, \text{if } i \text{ odd} \end{cases}, i \neq 1.$$

The recursive relation for G_{2n},

$G_{2n} = [M_1, M_{1.5}, M_2, ..., M_{2n}]$

can be expressed as

$G_k = k \cdot (k-1) \cdot G_{k-1}, k = (2n)div2$ and

$$G_{2n} = k \cdot \prod_{n=1}^{k-1} n^2 = k \cdot [(k-1)!]^2 = k! \cdot (k+1)!$$

If M, M' are two distinct permutations of {*1,2…,2n*} which differ at two positions, in order to transform M to M' it is necessary to transpose only two elements using minimal change for permutations. This is equivalent to saying that there exists an integer $i \in [2n]$ such that:

$$M'[j] = \begin{cases} M[j], j = i = 2n \\ M[j-2k], j = i + 2k, k = 1,2,...,2ndiv2 \\ M[j], j \neq i+2, j \neq i \end{cases}$$

The following recursive method can be defined, by repeating each permutation in the above G_{2n} list, *T=ndiv2* times, inserting another element (even/odd) after the last position of each M_i, and transposing the *T-1* permutations.

If $M_1 = [1,2]$ & $M_{1.5} = [1,2,3]$ then M_2 is obtained by inserting the digit 4 at the end of $M_{1.5}$ and taking two copies of this permutation:

1-2-3-4
1-2-3-4

Through a single transposition of the even digits 2 and 4 in the second permutation, $M_2 = \{[1,2,3,4],[1,4,3,2]\}$ is obtained.

The M_2 set, consisting of the permutations [1,2,3,4] & [1,4,3,2], is repeated twice and then the next digit, 5, is inserted at the end of each permutation.

1-2-3-4-5 1-4-3-2-5
1-2-3-4-5 1-4-3-2-5

By a single transposition of the odd digits 3 and 5 in both second cases of every pair of permutations, the results are respectively:

1-2-3-4-5 1-4-3-2-5
1-2-5-4-3 1-4-5-2-3

$M_{2.5} = \{[1,2,3,4,5],[1,2,5,4,3],[1,4,3,2,5],[1,4,5,2,3]\}$.

Finally, in the case of M_3 the algorithm inserts the digit 6 after the last position in every permutation of the above case and repeats each permutation of $M_{2.5}$ three times as shown in **Table 1**: This yields to all the permutations of M_3 (**Figure 1b**),

$$M_3 = \begin{Bmatrix} [1,2,3,4,5,6],[1,2,3,6,5,4],[1,6,3,4,5,2] \\ [1,2,5,4,3,6],[1,2,5,6,3,4],[1,6,5,4,3,2] \\ [1,4,3,2,5,6],[1,4,3,6,5,2],[1,6,3,2,5,4] \\ [1,4,5,2,3,6],[1,4,5,6,3,2],[1,6,5,2,3,4] \end{Bmatrix}$$

Table 1: M_3 Permutations

M_3 Permutations	
Permutations	Transposition of digits
1-2-3-4-5-6	
1-2-3-**4**-5-**6**	4,6
1-**2**-3-4-5-**6**	2,6
1-2-5-4-3-6	
1-2-5-**4**-3-**6**	4,6
1-**2**-5-4-3-**6**	2,6
1-4-3-2-5-6	
1-4-3-**2**-5-**6**	2,6
1-**4**-3-2-5-**6**	4,6
1-4-5-2-3-6	
1-4-5-**2**-3-**6**	2,6
1-**4**-5-2-3-**6**	4,6