ISO: Numeric Representation of Nucleic Acid Form

M. Leigh Fanning
Department of Computer Science
University of New Mexico
Albuquerque, New Mexico, 87131
leigh@unm.edu

Joanne Macdonald
Division of Experimental Therapeutics, Department of Medicine
Columbia University
New York, New York, 10032
jm2236@columbia.edu

Darko Stefanovic
Department of Computer Science
University of New Mexico
Albuquerque, New Mexico, 87131
darko@cs.unm.edu

ABSTRACT

We report a representation, ISO, that unambiguously and compactly describes patterns in nucleic acid secondary structure. ISO is equally expressive as other methodologies including dot-parenthesis notation, and various structure graph variations, for representation of well-formed structure patterns. ISO naturally expresses pseudoknot structures as well, without a change or extension in notation, an advantage not possible with commonly used representations. The numerical basis of ISO is readily amenable to development of mathematical evaluation rules, distance metrics, and further abstraction for either design or analysis purposes. In this way, ISO provides an easy mechanism for high-throughput in silico screening of sequence variants for architecting new synthetic systems, and better understanding of structure-function relationships and structure space in natural systems.

Categories and Subject Descriptors

B.1 [**B.1.4**]: [Languages and Compilers]; B.6 [**B.6.1**]: [Combinatorial Logic]; B.8 [**B.8.2**]: [Performance Analysis and Design Aids]; D.3 [**D.3.4**]: [Compilers]; D.4 [**D.4.8**]: [Modeling and Prediction]; E [**E.4**]: [Data Compaction and Compression]; I.2 [**I.2.4**]: [Representations]; J [**J.3**]: [Biology and Genetics]

General Terms

Algorithms, Design, Performance, Standardization, Verification

1. INTRODUCTION

Secondary structure elucidation of nucleic acids has been critical in understanding their functional behavior in biological systems. The inverse problem entails directing functional behavior by design in synthetic systems, where (deoxy)ribonucleic acids (DNA,RNA) can be used as a raw material for a variety of nanoscale applications. In our work, we devise molecular computing systems composed of short single-stranded oligonucleotides. In solution these oligonucleotides undergo hybridization, dissociation, and cleavage in a directed manner to serve as molecular scale computing primitives. Designing such a system requires arranging oligonucleotides to act as desired by individual selection of each particular base. These systems, similar to other molecular computing architectures, work without negative or positive feedback to regulate their actions. The absence of a control system places a certain burden on design to not only create intended effects, but to do so without the later opportunity to ameliorate noise.

Structural design requires a simple way to describe nucleic acid conformation. In describing DNA or RNA there is a hierarchy of data initiated with the list of bases *A*, *G*, *C*, and *T* for DNA or *A*, *G*, *C*, and *U* for RNA as the primary structure. Secondary structure denotes which bases are bound through Watson-Crick pairing of complementary bases $G-C$, and $A-T$ or $A-U$. Tertiary structure denotes spatial orientation, but is typically not yet included into large scale computational studies due to model complexity. Looking at the secondary structure for each molecule, and the change in free energy associated with folding to this structure, along with species concentrations and kinetic rates of reaction is sufficient to suggest and evaluate *in silico* prototype systems for laboratory verification. Since the number of choices of a single oligonucleotide is exponential in the length n of its string of bases ($\Sigma = \{A,C,G,(T,U)\}$, 4^n strings), and typically hundreds to millions of secondary structures per string may be evaluated as part of the design process, it has proven essential to use a numerical secondary structure abstraction, amenable to incorporation into a molecular compiler.

Previously developed secondary structure representations include the dot-parenthesis notation which was introduced in 1984 [7]. Thermodynamic design and modeling programs such as Vienna [6], Mfold [11], RNAsoft [1], and Nupack [21] all use the basic dot-parenthesis notation as either output for predicted conformations or input for determination of energy parameters associated with a desired conformation. This notation nominally uses a three-character alphabet $\{.,(,)\}$, where full stop ("dot") symbols indicate unpaired bases and matching parentheses indicate paired bases. Table 1 exemplifies this notation for the DNA oligonucleotide pictured in Figure 1(a). Strings with balanced parentheses describe structure patterns in which all hybridization regions are properly nested. The encoding is linear in the number of bases it abstracts and thus is not a particularly compact representation. Moreover, location information for interesting folding features (e.g. multibranching, stem-loops, or hairpins) can only be accomplished by overlaying a numeric index which is less efficient for molecular compiler incorporation.

Zuker and Sankoff [22], and Fontana [4], employ rooted trees to represent secondary structure where nodes denote base level information as shown in Figures 2b and 2c. Shapiro [15] also uses rooted trees, but introduces the notion of only describing structural

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
base	G	G	A	A	G	A	T	C	A	T	A	T	G	G	A	T	A	A	G	A	C	A
symbol	(	(	(	(	(	(	(	(	(	(	.	.	.	.	.	.	.	.	.	.	.	.
index	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
base	A	C	A	A	T	G	A	T	C	T	T	C	C	G	A	G	C	C	G	G	T	C
symbol	.	.	.	)	)	)	)	)	)	)	)	)	)	.	.	.	.	.	.	.	.	.
index	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
base	G	A	A	A	G	T	T	A	C	T	A											
symbol	.	.	.	.	.	.	.	.	.	.	.											

Table 1: Dot-Parenthesis representation for a 55 base DNA oligonucleotide, as pictured in Figure 1a.

features and their connectivity, while leaving off specific base information. Shapiro additionally shows a mapping from trees into nested feature lists in the style of the programming language LISP. Features include bulges (B), internal loops (I), multibranches (M) and hairpin loops (H), and thus one-dimensional lists are composed of their representative symbols in well-formed parenthesized list structures. Rather than rooted trees, Gan [5] uses planar graphs to describe structural features and how they are connected, and again leaves out specific base information. Ramlan [14] uses an extension to dot-parenthesis notation with a larger set of symbols to permit effective pictorial output. Oligonucleotide structure is represented by non-unique strings requiring reduction techniques to show equivalence, and human string parsing, outside the graphical display, is complex.

In each approach, the relevant questions are 1) how much structure information is conveyed? 2) how much is disregarded? 3) how compact is the representation? and 4) how can the abstraction be used effectively? Before the mid 1990s few groups were designing and building synthetic nucleic acid systems, hence application was solely directed at understanding structure-function relationships and formulating the sequence and structure space of nucleic acids, particularly RNA, found in natural systems. To explore the space of secondary structure, distance metrics between structures and structure alignment techniques are required. The study of natural systems is ongoing, therefore these needs persist, but they are now augmented by design challenges in building synthetic systems.

To address these needs, we invented and have been actively using a novel structure representatation, *ISO*. We present it here as useful alternative to other previously developed notations, and illustrate its capability in the design process. Advantages of ISO over other representations include the following:

- The ability to express, without a change in notation, pseudoknot structure wherein hybridization regions are not properly nested.
- The ability to characterize synthetic nucleic acid systems, and their design, in numerical terms amenable to molecular compiler incorporation.
- The ability to characterize the space of naturally occurring nucleic acid structure as operators acting on operands.
- The notation is compact and simple, yet maximal structural information is retained.

In the remaining sections we define ISO, show its expressiveness for any arbitrary folding pattern including pseudoknots, demonstrate applicability towards system design, and provide mappings to several alternative representations.

2. ISO

The high information content of the dot-parenthesis notation, coupled with the need to efficiently evaluate large numbers of DNA oligonucleotide conformations, led us to consider something equally straightforward, but more amenable to writing classification rules which could abstractly capture years of experimentalist knowledge and keen laboratory insight. With this in mind, we examined how structural information is used, and determined that a numeric representation would accelerate synthetic nucleic acid system design, as well as serve towards sequencing, structure-function, and structure space studies.

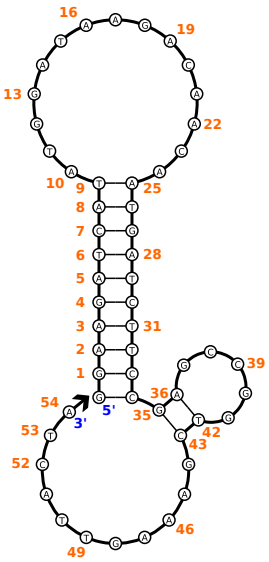
ISO notation describes nucleic acid secondary structure as a list of triples (***i**ndex*, ***s**tem*, ***o**pening)*, where each triple defines a distinct hybridization region within a single nucleic acid oligonucleotide, or between multiple oligonucleotides making up a complex.

Definition Let $P = \{p_0, p_1, \dots\}$ be a set of n nucleotide strings, drawn from $\Sigma = \{A, C, G, T, U\}$, and $d \notin \Sigma$ be a neutral spacer symbol. Form concatenated string c by ordering $5'$ to $3'$ all strings $p_i \in P$, separating each two p_i by d such that $c = p_0 d p_1 d \dots d p_{n-1}$. Let t be a list of m triples, $t = [(i,s,o)_0, (i,s,o)_1, \dots, (i,s,o)_{m-1}]$. t is a unique representation of secondary structure features in c where for each feature:

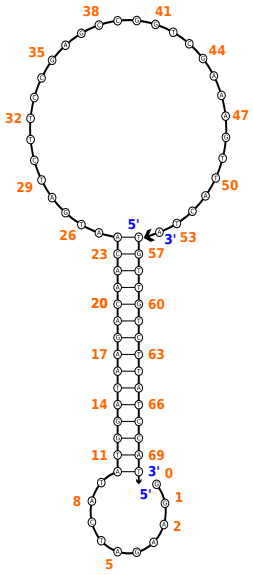
1. i defines the (zero-based) indexing location relative to the p_0 $5'$ end.
2. s defines the length of binding stem.
3. o defines the opening enclosed by s, equal to the number of bases, paired or unpaired, which are intermediate between the last opening base and first closing base of this feature.

Figure 1a exemplifies this notation for an individual oligonucleotide sequence designed to fold into a structure with two stem-loops, one large and one small. We compare and show the ISO and dot-parenthesis notation for this structure as:

- $[(0,10,15),(32,2,5)]$
- ((((((((((.....))))))))))((.....)).....



(a) Single DNA oligonucleotide.



(b) Two DNA oligonucleotides forming a complex.

Figure 1: (**a**) Secondary structure for a 55 base DNA sequence. The first feature starts at base 0, with a binding stem comprising 10 base pairs, encompassing a loop opening of 15. The second feature starts at base 35, has a 2 base pair binding stem, and encompasses a smaller loop opening of 5. (**b**) Hybridization structure between two DNA sequences with lengths of 55 and 15 bases. The single feature starts at base 10, with a binding stem comprising 15 base pairs, encompassing an opening of 31, including the spacer (not shown).

Figure 1b exemplifies a complex of two oligonucleotides designed to bind together. For these cases where multiple oligonucleotides are concatenated, spacer characters are counted as part of the indexing scheme. The length of the sequence, or sequences and spacer characters, is not captured in this representation and must be supplied separately when needed. For comparison, the corresponding ISO and dot-parenthesis notation for this structure is:

- $[(10,15,31)]$
- .....((((((((((((((.....+))))))))))))))

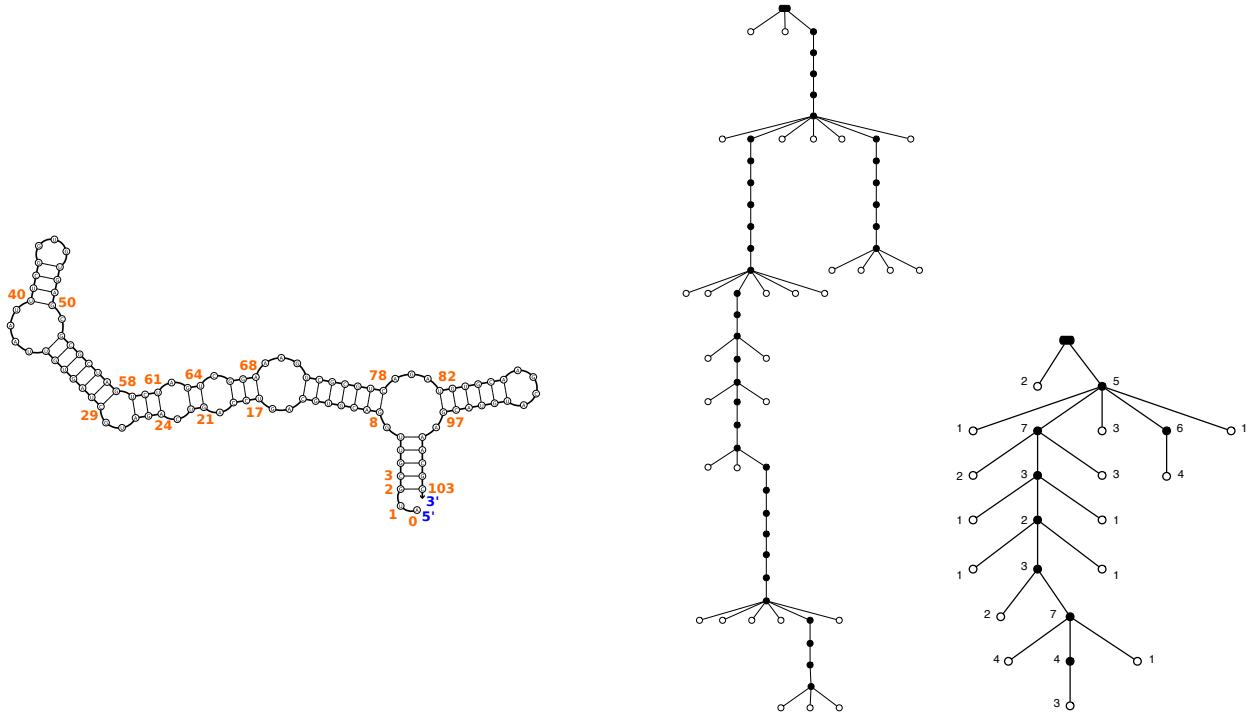
3. ISO EXPRESSIVE POWER

Nucleic acids may naturally, or as synthetically directed, prefer to fold into a variety of motifs including bulges, internal loops, hairpins, stem-loops and multibranches. ISO expresses all structure information exactly for each of these forms by virtue of relationships between the triples. Consider a structure with m features, $[(i,s,o)_0,(i,s,o)_1,\ldots,(i,s,o)_{m-1}]$. A motif anchored as feature j, $j \in [0,m-1]$ is identified in the following ways. In each of these we note further that not only can we recognize existence of a specific feature, and locate it precisely, we can also infer the size exactly via further simple arithmetic over the triples defining the feature.

1. **Bulges.** A bulge is one or more unpaired bases on one side of two stem regions. A bulge is recognized within the list of triples where features $(i,s,o)_j$ and $(i,s,o)_{j+1}$ satisfy either of the following, but not both:
   - $i_{j+1} - (i_j + s_j) > 0,\ (i_j + s_j + o_j) - (i_{j+1} + 2s_{j+1} + o_{j+1}) = 0$
   - $i_{j+1} - (i_j + s_j) = 0,\ (i_j + s_j + o_j) - (i_{j+1} + 2s_{j+1} + o_{j+1}) > 0$

   We see an example of a bulge in Figure 2(a), in triples 4 ($(24,3,32)$) and 5 ($(29,7,16)$), since $i_5 - (i_4 + s_4) = 29 - (24+3) > 0$ and $(i_4 + s_4 + o_4) - (i_5 + 2i_5 + o_5) = (24+3+32) - (29+14+16) = 0$.
2. **Internal loops.** An internal loop is formed by unpaired open regions surrounded by exactly two stems, where at least one of the unpaired bases must occur on both sides. An internal loop is recognized within the list of triples where features $(i,s,o)_j$ and $(i,s,o)_{j+1}$ satisfy:
   - $i_{j+1} - (i_j + s_j) > 0,\ (i_j + s_j + o_j) - (i_{j+1} + 2s_{j+1} + o_{j+1}) > 0$

   One internal loop example in Figure 2(a) is found between the stems represented by triples 1 ($(8,7,57)$) and 2 ($(17,3,46)$), since $i_5 - (i_4 + s_4) = 17 - (8+7) > 0$ and $(i_4 + s_4 + o_4) - (i_5 + 2s_5 + o_5) = (8+7+57) - (17+6+46) > 0$.
3. **Hairpins.** Hairpins are terminal stems with no unpaired bases intervening. Hairpins are recognized as a triple with a size zero opening:
   - $(i,s,0)_j$
4. **Stem-loops.** Stem-loops are hairpins with at least one unpaired base between the opening and closing bases of the stem. Stem-loops are recognized as a triple:



(a) Graphical folding. (b) Structure tree. (c) Condensed structure tree.

Figure 2: **(a)** 105 base RNA segment with two stem-loops, five internal loops and one three-way multi-branch feature. The ISO representation, starting from the marked 5′ end, is $[(2,5,92),(8,7,57),(17,3,46),(21,2,40),(24,3,32),(29,7,16),(40,4,3),(82,6,4)]$. **(b)** The corresponding structure tree representation. **(c)** The structure tree representation condensed further.

- $(i,s,>0)_j$

One example stem-loop is seen in Figure 2(a) as the last triple $((82,6,4))$, where a binding stem of six base pairs surrounds four unpaired bases.

5. ***R*-way multibranches.** An *r*-way multibranch is an internal loop formed by *r* surrounding stems. An *r*-way multibranch is recognized within the list of triples where exactly $r-1$ features $(i,s,o)_{j+1},\ldots,(i,s,o)_{j+r-2}$ are enclosed by feature $(i,s,o)_j$, but not enclosed by each other:

   - $\forall i_k, k \geq j+1$, $i_j+s_j < i_k < i_j+s_j+o_j$, $i_k+2s_k+o_k > i_{k-1}+2s_{k-1}+o_{k-1}$

   The first constraint tracks binding openings and stipulates that each triple defining a multibranch stem follows the initiating 5'-most stem, and is completely enclosed by the opening and closing bindings of this stem. The second constraint tracks binding closings and stipulates that subsequent stems *not* be enclosed by any previous defining one. For example, in Figure 2(a), triple 0 $((2,5,92))$ initiates a 3-way branch, and triples 1-7 are enclosed by this triple. However, triples 2-6 are excluded since each of their extents is enclosed by triple 1 $((8,7,57))$, and therefore they fail to satisfy the second constraint. Hence, the 3-way branch is represented by triples 0, 1, and 7, equal to sublist $[(2,5,92),(8,7,57),(82,6,4)]$.

A different, important form is a pseudoknot, which typically is harder to express. *Pseudoknots* are patterns containing regions of intercalated hybridization.

Definition Consider the set of base pair indices relative to the 5′ end. For any two pairing indices *a*,*b* and *c*,*d* hybridized as $a-b$ and $c-d$ and where *a* and *c* are opening base indices and *b* and *d* mark their respective closing bases, if $a<c<b<d$, then the opening of *c*-*d* occurs before the closing of *a*-*b* and these pairs are said to be pseudoknotted.

RNA found in nature folds into pseudoknots [16] as a result of additional stacking plane hydrogen bond opportunities which yield stability benefits despite the asymmetric and jumbled appearance. An example pseudoknot is shown in Figure 3. RNA has evolved to use pseudoknots for a variety of cellular functions including self-cleavage of ribozymes, frameshifting the coding regions for viruses during translation, processing activity of telomerases, and autoregulation of viral gene expression [2]. Dot-parenthesis notation requires the addition of new symbols to distinguish between pairing regions, typically square brackets ($[,]$) or curly braces ($\{,\}$). As an example, **((.....))**, and **()..()** are properly nested, while **(([[..))]]** is pseudoknotted. Each new intercalated binding region requires a new matching symbol, or some method of indexing. Suggestions have also been made to show each distinct intercalated binding region with a different color. Pseudoknots cannot be represented by rooted trees or planar graphs. However, they are naturally expressed in ISO. We add pseudoknots to our list of expressible features, again assuming a structure represented by ISO as follows:

6. **Pseudoknots.** A pseudoknot is intercalated (non-nested) structure. A pseudoknot is is recognized in structure *S*, represented by ISO $S=[\ldots,(i,s,o)_j,\ldots(i,s,o)_k,\ldots]$, $k \geq j+1$

where features $(i,s,o)_j$ and $(i,s,o)_k$ satisfy:

- $i_k < i_j + s_j + o_j$, $i_k + s_k + o_k > i_j + 2s_j + o_j$

The first constraint places the opening bases of feature $(i,s,o)_k$ within the unpaired region of feature $(i,s,o)_j$, before the location of its closing bases. The second constraint places the corresponding closing bases of $(i,s,o)_k$ outside feature $(i,s,o)_j$, hence the features are not nested. The intercalation may occur for successive feature triples, or any two feature triples separated within the ISO list. Counting all such pairs of triples where this test holds yields the degree of knots. We see an example in Figure 3 below, where the two triples representing the structure satisfy the pseudoknot test.

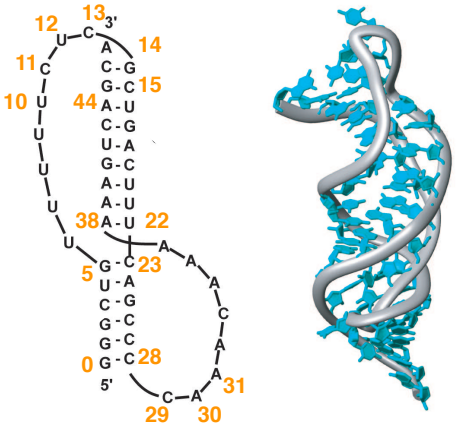


Figure 3: Human telomerase (hTR) pseudoknot structure located at the 5' end of the 451 base RNA [16]. The 49 base pseudoknot sequence segment is GGGCUGUUUUUCUCGCUGACUUUCAGCCCCAAACAAAAAAGUC which folds into a H-type motif with two stems and two loop regions. We describe this pseudoknot as $[(0,6,17),(14,9,14)]$.

4. EQUIVALENCE TO ALTERNATIVE REPRESENTATIONS

ISO is equivalent to several alternative representations. We separately show this for dot-parenthesis notation and two tree variants.

4.1 Equivalence to dot-parenthesis notation

ISO and dot-parenthesis notation are equivalent by inspection, except for sequence length. ISO indexing (i), stem length (s), and loop opening (o), for each hybridization region, are directly inferred from dot-parenthesis notation by starting at the $5'$ end of a structure string and noting the index location of each opening base pair, the number of base pairs before a loop opening, and the intervening unpaired bases encountered before the first closing base pair is reached. Sequence length is implicit in dot-parenthesis simply by counting the total number of structure characters.

We show two algorithms to transform dot-parenthesis notation into ISO and the reverse direction, from ISO into dot-parenthesis. First, to transform dot-parenthesis notation into ISO, a linear scan of the input string yields two lists of open and close parentheses. Repeating until list *stackclose* is empty, we examine list *stackopen* from the end forward to find the corresponding opening parenthesis. From this location, the largest stem-loop possible is built and saved as a complete ISO triple. Participating *stackopen* and *stackclose* elements are removed as each ISO triple completes. The algorithm runs in $O(n)$ time for a length n dot-parenthesis string.

```
Dot-Parenthesis-to-ISO(oligos,triples)
1   stackopen ← [ ]
2   stackclose ← [ ]
3   triples ← [ ]
4   idx ← 0
5   while idx < len(oligo)
6       if oligo[idx] ≡ "("
7           append(stackopen, input[idx])
8       if oligo[idx] ≡ ")"
9           append(stackclose, input[idx])
10  if len(stackopen) < 1
11      return triples
12  while len(stackclose) > 0
13      anchor_close_value ← stackclose[0]
14      open_idx ← len(stackopen) - 1
15      while anchor_close_value < stackopen[open_idx]
16          open_idx ← open_idx − 1
17      open_killset ← [stackopen[open_idx]]
18      close_killset ← [stackclose[0]]
19      previous_open_val ← stackopen[open_idx]
20      previous_close_val ← stackclose[0]
21      open_idx ← open_idx − 1
22      close_idx ← 1
23      i ← stackopen[open_idx]
24      s ← 1
25      o ← anchor_close_value − stackopen[open_idx] − 1
26      while open_idx > −1 ∧ close_idx < len(stackclose)
27          if stackclose[close_idx] − previous_close_val ≡ 1 ∧
28          previous_open_val − stackopen[open_idx] ≡ 1
29              append(close_killset, stackclose[close_idx])
30              append(open_killset, stackopen[open_idx])
31              previous_close_val ← stackclose[close_idx]
32              previous_open_val ← stackopen[open_idx]
33              open_idx ← open_idx − 1
34              close_idx ← close_idx + 1
35              i ← i − 1
36              s ← s + 1
37          else
38                  break
39          append(triples, (i,s,o))
40          foreach element ∈ open_killset
41              delete(stackopen, element)
42          foreach element ∈ close_killset
43              delete(stackclose, element)
44  return sorted(triples)
```

To transform an ISO list into dot-parenthesis notation we require additional length information input. We form a list of full stop (dot) characters to the input length and replace each with either open or close parenthesis symbols based on examination of each ISO structure in the list. We convert the list into a string as output. The algorithm runs in $O(n)$ time for a length n dot-parenthesis string.

```
ISO-to-Dot-Parenthesis(triples,length)
1   dpstring ← [ ]
```

```
2   dpidx ← 0
3   while dpidx < length
4           append(dpstring, ".")
5           dpidx ← dpidx + 1
6   for iso ∈ triples
7           sdx ← 0
8           while sdx < iso[1]
9                   dpstring[iso[0] + sdx] ← "("
10                  dpstring[iso[0] + iso[1] + iso[2] + sdx] ← ")"
11                  sdx ← sdx + 1

12  return string(dpstring)
```

4.2 Equivalence to tree representations

Since dot-parenthesis notation and tree representations are equivalent, ISO is equivalent to tree representations as well. We discuss tree schemes further, and a direct mapping to two different tree variations.

Following the graph-based approach of Waterman [19] used to enumerate the possible structures for a length *n* oligonucleotide, Zuker and Sankoff used tree representations to depict secondary structure in their review of the structure prediction problem [22]. A structure tree (Figure 2b) is rooted by a benign, non-participating vertex representing the 5' end of a sequence. Remaining vertices are mapped to two states, open and closed, where open represents a single unpaired base and closed represents a base pair, encountered when moving towards the 3' sequence end. Parent-child edge relationships depict the notion of nesting with each downward edge. Closed connected vertices represent moving symmetrically inward along a stem towards the enclosed unpaired loop bases which are all leaves. Hairpins are recognized as closed leaf vertices.

Motivated by characterizing the structure space of all RNA conformations, Fontana updated this representation [4] by condensing parts of a structure tree in two ways. First, the contiguous edges of closed vertices are collapsed into a single edge, with parent vertex labels denoting the number of condensed contiguous edges and therefore clearly representing the length of a stem formed by the base pairs at that location. Second, unpaired bases represented by open vertices are collected into a single open vertex and labeled with the count of unpaired bases at that location. The resulting trees (Figure 2c) have no degree two vertices and are *homeomorphically irreducible trees* (HITs). The HIT closed vertex labels are equivalent to ISO *s* values. The sum of all child vertex labels for any closed vertex parent is equivalent to ISO *o* values. Since HITs are a reduction of structure trees, we find the same equivalences between ISO and structure trees.

5. DEOXYRIBOZYME DESIGN USING ISO

Our approach to computing based on synthetic nucleic acids relies on hybridization between complementary oligonucleotides, and the ability to cleave one oligonucleotide into two using deoxyribozymes. Deoxyribozymes are used as logic gates, along with single stranded oligonucleotides as inputs and outputs [17, 18, 10, 13]. Logic gates are designed to fold into one or more stem-loop conformations, (e.g. Figure 1a), where each loop serves as the recognition region for its intended complementary input (e.g. Figure 1b). Logic gate primitives are able to execute the basic Boolean connectives *not*, *and*, and *or*, as well as several extended relations between multiple inputs. One or more gates together with their inputs solve problems as instances of combinatorial logic in solution.

Deoxyribozyme system design must ensure complementary oligonucleotides where hybridization is desired, and minimize unwanted interactions between oligonucleotides present in solution, but not intended to work together. To this end, thermodynamic modeling programs allow prototype evaluation of various sequence selections for each of the constituent elements either individually, or in various combinations. The small number of elements is not indicative of system complexity. For example, it is not the case that a single gate, input, and substrate sequence only yield $\binom{3}{2}$ pairs to evaluate together and only 3 sequences to evaluate alone. Modeling instead considers systems as large scale ensembles distributed over a set of folding states. Some of these states will be advantageous and entail deoxyribozyme gate oligonucleotides conforming perfectly to designed folds, hybridizing correctly to input oligonucleotides, and cleaving substrates exactly. Others will be deleterious to some degree and therefore contribute to system noise and possible failure to yield adequate output product. A more careful design entails considering an entire range of structures exposed as possible folding states, and their associated predicted free energies to determine the overall energy landscape of competing structures.

The combinatorial nature of systems requires fast evaluation of large numbers of secondary structures. To this end, we encode each modeled structure into ISO representation. Equations over ISO instances are formulated as structure design specification rules, allowing comparison of each ISO to multiple predefined feature patterns and determination of an overall weighted score based on quantified similarity each feature to its design pattern. Separate evaluations occur for each deoxyribozyme gate motif alone, the gate together with its input as a hybridized complex, and the gate together with its substrate as a hybridized complex. In each regime, we have encoded design specifications for secondary structure by decomposing the structure into multiple features of interest, each of which is scored independently on a scale from 0.0 (failure) to 1.0 (perfect). We show a representative subset of scoring rules in Table 2 for one deoxyribozyme gate design which recognizes a single input, and therefore has a single stem-loop as shown in Figure 1(a). The location information provided by the index (i), along with the sizes of the stem (s) and opening (o), allows pattern recognition of a modeling result against specifications. Together, these steps link the energy, form and desired function into a single expectation of desired behavior. Form and function mapping is not arbitrary, and instead abstracts long-term laboratory results and experience from previous studies. The ISO output and scoring for a set of folding states for one YES gate, generated by the Nupack thermodynamic modeling code [21], is presented in Table 3. Using the scoring rules, we can rapidly evaluate thousands of YES gate variants, and provide a ranked output in terms of predicted utility.

6. DISCUSSION

We have shown ISO as a novel, unambiguous, representation for secondary structure. We focus on the single most critical aspect of working with nucleic acids, precisely showing where bindings are occurring. This focuses attention onto the positive space of oligonucleotides, yet it also reveals the negative space, the areas of strands which are unbound, and therefore available for subsequent bindings. As shown here, this is equivalent to several other schemes, however ISO is more informative, more compact, and more expressive than any other scheme. ISO is amenable to use as part of nucleic acid functionality standards development and exchange of information, such as the RNAML syntax [20].

ISO was motivated by different reasons than development of other

Rule	Feature $(i,s,o)_j$	Feature(s) $(i,s,o)_k, k \geq j+1$	Score
stem-1	$i_j = 0, s_j > 9$		1.00
stem-2	$i_j = 1, s_j > 8$		0.90
stem-3	$i_j = 0, s_j = 9$		0.90
loop-1	$o_j > 14$	$i_k > i_j + 2s_j + 15$	1.00
loop-2	$o_j = 13, i+s = 11$	$i_k > i_j + 2s_j + 15$	0.80
loop-3	$o_j = 11, i_j + s_j = 12$	$i_k > i_j + 2s_j + 15$	0.60
loop-4	$o_j < 11, i_j + s_j > 12$	$i_k > i_j + 2s_j + 15$	0.00

Table 2: Selected set of design specifications for deoxyribozyme 8.17.1 Yes Gate [13]. Structure $S = [(i,s,o)_0, \ldots, (i,s,o)_{m-1}]$, with m total features, is decomposed into required and incidental successive features $j,k \in [0, m-1]$. Feature $(i,s,o)_j$ must occur, whereas additional features may or may not be present. Stem rules 1-3 check for a stem with perfect form, shifted by one base, or short by one base pair, respectively. Loop rules 1-4 check for a clean loop, a loop encroached by the stem by one base pair, a loop encroached by the stem by two base pairs, a loop encroached by more than two base pairs, respectively. Other rules, not shown, check for further variations.

ISO	Probability	Stem Score	Stem Expectation	Loop Score	Loop Expectation
[(0,10,15)]	0.20	1.0	0.20	1.0	0.20
[(0,10,15),(14,2,4)]	0.08	1.0	0.08	0.8	0.07
[(0,9,17)]	0.06	0.9	0.05	1.0	0.06
[(1,9,15),(34,3,5)]	0.06	0.9	0.05	1.0	0.06
[(0,10,15),(14,2,6)]	0.03	1.0	0.03	0.8	0.03
[(0,10,15),(43,1,4)]	0.03	1.0	0.03	1.0	0.03
[(0,10,15),(35,2,5)]	0.03	1.0	0.03	1.0	0.03
[(1,9,15),(14,2,4),(34,3,5)]	0.02	0.9	0.02	0.8	0.02
[(0,10,15),(14,1,4)]	0.02	1.0	0.02	0.9	0.02
[(1,8,17),(34,3,5)]	0.02	0.0	0.00	1.0	0.02
[(0,10,15),(14,2,4),(43,1,4)]	0.01	1.0	0.01	0.8	0.01
[(0,10,15),(14,2,4),(35,2,5)]	0.01	1.0	0.01	0.8	0.01
[(0,10,15),(14,1,6)]	0.01	1.0	0.01	0.9	0.01
[(1,9,15),(14,2,6),(34,3,5)]	0.01	0.9	0.01	0.8	0.01
[(0,10,15),(41,2,8)]	0.01	1.0	0.01	1.0	0.01

Table 3: Deoxyribozyme 8.17.1 Yes Gate example evaluation results for 15 most probable variants of a single gate sequence. Probability of occurrence computed as a function of predicted free energy and the partition function, with scores based on rules displayed in Table 2, and computed expectation as the product of score and probability.

representation schemes, yet the importance of secondary structure representation remains consistent. Cataloging and classifying secondary structure in terms of feature connections is an active research area [3, 8, 9, 12]. ISO keeps feature information location and size, therefore in addition to connection determination, distinction can be made for feature extents rather than abstracted away. By virtue of using numbers to describe what is ultimately hydrogen bonding between molecular chains, we have greater ability to cast the understanding of nucleic acid structure space as a pattern recognition problem. Indeed, this understanding can be done using mathematical expressions, combining ISO and simple arithmetic relations, without resorting to graph theory. Moreover, it is straightforward to write a distance metric between two ISO instances, or show structure alignment. In both cases, $O(nm)$ pairwise triple comparisons of some two ISO instances, α of length n, and β of length m, can be computed to find pattern differences and matchings.

For synthetic system design, secondary structure in molecular computing systems serves as a compiled-to instruction. Direct parallels between electronic and molecular computing are not straightforward, but in this case one parallel can be observed. Just as architecture specific instructions in electronic computers direct movement of data up and down the memory hierarchy, and execution of operations on data in a very low-level manner, nucleic acids also store encoded data, and interactions between them serve to execute operations upon this data. These interactions can only occur through the physical effects of hybridization and cleavage, as specified directly by their forms. Form instructs function. Orchestrating tens to thousands of simultaneous forms demands instructions which are easy to incorporate into a molecular compiler. A numeric representation for instructions enables compiler construction encompassing both rule-sets as demonstrated here, as well as optimization where structure specifications can be incrementally modified in a generate-and-test scheme within the combinatorial sequence space of participating elements.

7. ACKNOWLEDGMENTS

Annotated drawings in Figures 2 and 1 were created using the Nupack nucleic acid modeling tool suite [21]. Annotated Figure 3 is from Reference [16]. RNA and structure trees in Figure 2 are from Reference [4]. The material is based upon work supported by the National Science Foundation under NSF Grants 0829881 and 0829793.

8. REFERENCES

[1] M. Andronescu, R. Aguirre-Hernandez, A. Condon, and H. Hoos, *RNAsoft: a suite of RNA secondary structure prediction and design software tools*, Nucleic Acids Research **31** (2003), no. 13, 3414–3422.

[2] I. Brierley, S. Pennell, and R. Gilbert, *Viral RNA pseudoknots: versatile motifs in gene expression and replication*, Nature Reviews Microbiology **5** (2007), 598–610.

[3] D. Fera, N. Kim, N. Shiffeldrim, J. Zorn, U. Laserson, H. Gan, and T. Schlick, *RAG: RNA-as-graphs web resource*, BMC Bioinformatics **5** (2004), 88.

[4] W. Fontana, D. Konings, P. Stadler, and P. Schuster, *Statistics of RNA secondary structures*, Biopolymers **33** (1993), 1389–1404.

[5] H. Gan, S. Pasquali, and T. Schlick, *Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design*, Nucleic Acids Research **31** (2003), no. 11, 2926–2943.

[6] I. Hofacker, W. Fontana, P. Stadler, M. Bonhoeffer, M. Tacker, and P. Schuster, *Fast folding and comparison of RNA secondary structures*, Chemical Monthly **125** (1994), 167–188.

[7] P. Hogeweg and B. Hesper, *Energy directed folding of RNA sequences*, Nucleic Acids Research **12** (1984), no. 1, 67–74.

[8] Y. Karklin, R. Meraz, and S. Holbrook, *Classification of non-coding RNA using graph representations of secondary structure*, Pacific Symposium on Biocomputing (2005), 4–15.

[9] M. Khaladkar, V. Bellofatto, J. Wang, B. Tian, and B. Shapiro, *RADAR: a web server for RNA data analysis and research*.

[10] J. Macdonald, Y. Li, M. Sutovic, H. Lederman, K. Pendri, W. Lu, B. Andrews, D. Stefanovic, and M. Stojanovic, *Medium scale integration of molecular logic gates in an automaton*, Nano Letters **6** (2006), no. 11, 2598–2603.

[11] N. Markham and M. Zuker, *DINAMelt web server for nucleic acid melting prediction*, Nucleic Acids Research **33** (2005), W577–W581.

[12] G. Pavesi, G. Mauri, M. Stefani, and G. Pesole, *RNAprofile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences*, Nucleic Acids Research **32** (2004), no. 10, 3258–3269.

[13] R. Pei, E. Matamoros, M. Liu, D. Stefanovic, and M. Stojanovic, *Training a molecular automaton to play a game*, Nature Nanotechnology **5** (2010), 773–777.

[14] E. Ramlan and K. Zauner, *An extended dot-bracket-notation for functional nucleic acids*, International Workshop on Computing with Biomolecules (2008), 75–86.

[15] B. Shapiro, *An algorithm for comparing multiple RNA secondary structures*, Bioinformatics **4** (1988), no. 3, 387–393.

[16] D. Staple and S. Butcher, *Pseudoknots: RNA structures with diverse functions*, PLoS Biology **3** (2005), no. 6.

[17] M. Stojanovic, T. Mitchell, and D. Stefanovic, *Deoxyribozyme-based logic gates*, Journal of the American Chemical Society **124** (2002), no. 14, 3555–3561.

[18] M. Stojanovic and D. Stefanovic, *A deoxyribozyme-based molecular automaton*, Nature Biotechnology **21** (2003), no. 9, 1069–1074.

[19] M. Waterman, *Secondary structure of single-stranded nucleic acids*, Studies on Foundations and Combinatorics, Advances in Mathematics Supplementary Studies (1978), 167–212.

[20] A. Waugh, P. Gendron, R. Altman, J. Brown, D. Case, D. Gautheret, S. Harvey, N. Leontis, J. Westbrook, E. Westhof, M. Zuker, and F. Major, *RNAML: A standard syntax for exchanging RNA information*, RNA **8** (2002), no. 6, 707–717.

[21] J. Zadeh, C. Steenberg, J. Bois, B. Wolfe, M. Pierce, A. Khan, R. Dirks, and N. Pierce, *NUPACK: analysis and design of nucleic acid systems*, Journal of Computational Chemistry **32** (2011), 179–173.

[22] M. Zuker and D. Sankoff, *RNA secondary structures and their prediction*, Bulletin of Mathematical Biology **46** (1984), no. 4, 591–621.