Automated Alignment of RNA Sequences to Pseudoknotted Structures

Jack E. Tabaska and Gary D. Stormo
Dept. of Molecular, Cellular, and Developmental Biology
University of Colorado
Boulder, CO 80309–0347
jtabaska@ural.colorado.edu
stormo@ural.colorado.edu

Abstract

Seq7 is a new program for generating multiple structure-based alignments of RNA sequences. By using a variant of Dijkstra's algorithm to find the shortest path through a specially constructed graph, *Seq7* is able to align RNA sequences to pseudoknotted structures in polynomial time. In this paper, we describe the operation of *Seq7* and demonstrate the program's abilities. We also describe the use of *Seq7* in an Expectation-Maximization procedure that automates the process of structural modeling and alignment of RNA sequences.

Introduction

Computer methods for performing RNA sequence alignments have traditionally relied on sequence similarity to detect homologies within a family of RNAs. It is widely recognized, however, that evolution selects for conservation of an RNA's three-dimensional conformation more so than for preservation of nucleotide sequence. Sequence similarity-based RNA alignment methods therefore do not take into account some of the most important information available on the sequences they are aligning.

Recently, a number of methods have been developed which incorporate higher-order structural information on RNAs into sequence alignments (Eddy & Durbin 1994; Gautheret et al. 1990; Kim et al. 1996; Sakakibara et al. 1994), with good results. These methods, however, are limited by the fact that the general problem of aligning a sequence to a structure, including nonlocal interactions and variable-length gaps, is NP-hard (Lathrop 1994). Any computationally efficient program that attempts to align RNA sequences to a structure must therefore sacrifice generality or optimality. For instance, approaches based on stochastic context-free grammars (Eddy & Durbin 1994; Sakakibara et al. 1994) can find optimal structural alignments, but only to planar structures. Other approaches based on string matching algorithms (Gautheret et al. 1990) or simulated annealing (Kim et al. 1996) are able to align sequences to pseudoknotted structures, but with restrictions on the size or placement of gaps.

Since it seems unlikely that P = NP, it appears that the best we can do to solve an RNA structural alignment problem is to make use of different approaches with overlapping strengths and weaknesses. For this reason, we have developed a new structural alignment program called *Seq7*. *Seq7* takes a novel graph-theoretical approach to performing structure-based multiple RNA sequence alignments, allowing the program to find near-optimal alignments without gap restrictions to nonplanar structures in polynomial time. We discuss here the operation of *Seq7* and demonstrate its ability to align RNA sequences to pseudoknotted structures. We also show how *Seq7* may be used as part of an Expectation-Maximization method for elucidating the higher-order structure of RNAs and producing structure-based alignments in the absence of preexisting structural information.

Methods

Seq7

Model structures. *Seq7* constructs multiple RNA sequence alignments by aligning a set of related RNA sequences to a common model structure. The model structure is considered to be a pseudo-RNA sequence which is prototypical of all of the sequences to which it is to be aligned, similar to a profile. The model consists of a number of model positions, each of which includes information on the structural milieu, base identity, and insertion/deletion propensities of one base of this pseudosequence. This latter property of *Seq7* model structures allows for positionally variable gap penalties.

The structural information contained in a model position is in the form of a pairing partner specification, that is, the model position with which a particular base interacts in the folded RNA. Unpaired bases are simply noted as not having any pairing partner at all. Describing RNA structures in this manner is flexible enough to allow representation of pseudoknotted structures, and potentially other types of tertiary interactions.

Nucleotide sequence information in the model is expressed as positional base preferences, which may be thought of as *a priori* estimates of base representation at each model position in the final alignment. Model positions involved in base pairing interactions also contain information about the preferred identity of their pairing partners.

Alignment graphs. At the heart of *Seq7* is Dijkstra's shortest path algorithm (Dijkstra 1959). This algorithm finds the shortest route from one point to another through a network of interconnecting pathways, modeled as a graph. Using Dijkstra's algorithm, the shortest path through a graph can be found in polynomial time.

To apply Dijkstra's algorithm to the structural alignment problem, it is necessary to construct a graph in which every possible alignment of an RNA sequence to a model structure is represented by a path. The lengths of these paths must be inversely related to how good the corresponding alignment is: short paths represent good alignments of the sequence to the structure, and long paths represent poor alignments. Figure 1a depicts such a graph for a situation in which the model structure contains no base pairs, for the simpler problem of aligning two sequences. The graph contains a grid of vertices. A vertex at (x, y) coordinates (i, j) represents the alignment of base j in the test sequence with model position i. Each vertex receives a weight based on how well the base in the sequence matches the base probabilities specified by the model description at that position. Good matches receive a low weight, while poor matches receive higher weights.

The edges of the alignment graph represent the gaps that must be introduced to move from one alignment position to another. For clarity, the edges emanating from a single vertex in the alignment graph are shown in figure 1b. The edges that run parallel to the diagonal of the vertex grid, i.e. from (i, j) to $(i + 1, j + 1)$, represent situations where no gap needs to be introduced into the alignment. These edges receive a weight of zero. Other edges connect each vertex (i, j) with $(i+\Delta x, j+1)$ or with $(i+1, j+\Delta y)$, where $\Delta x, \Delta y > 1$. These edges represent, respectively, deletions and insertions in the sequence with respect to the model, and are given weights that increase in proportion to the gap

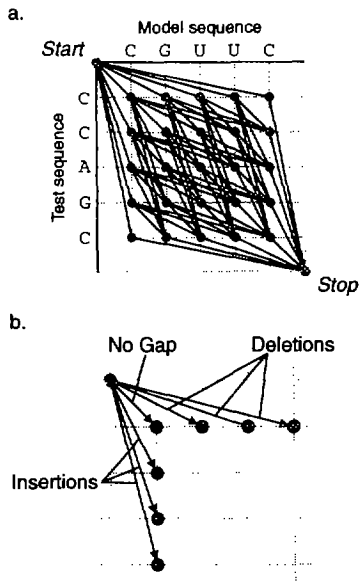


Figure 1: (a) Graph for alignment of two sequences. (b) Pattern of edges emanating from a single vertex in the alignment graph. Note that the edges in (a) are actually directed as shown in (b), but the arrowheads have been omitted for clarity.

length x. Note that edges representing every possible insertion or deletion following a given alignment position exist, thus allowing for variable-length gaps.

There are also two vertices outside of the grid, labeled *Start* and *Stop* (fig. 1a). These vertices do not represent any physical feature of the model or the sequence, but are simply there to provide Dijkstra's algorithm with definite starting and stopping points. *Start* and *Stop* are connected to the rest of the graph by edges that represent 5′ and 3′ terminal gaps, and which are weighted according to gap length.

If we define the length of a path from *Start* to *Stop* on this graph as the sum of the weights of the vertices encountered and edges traversed along the path, Dijkstra's shortest path algorithm will find the alignment containing the lowest-scoring combination of mismatched bases and gaps — that is to say, the best alignment.

To include base pairing information in such an alignment graph, the vertex grid must be made three-dimensional so as to allow two bases to be simultaneously aligned with each base pair in the model structure. A vertex at (x, y, z) coordinates (i, j, k) then stands for the alignment of base pair $j : k$ with the base pair specified by model position i and i's pairing partner, i^*. Vertices are weighted according to how well base pair $j : k$ matches the base pair preferences of model position i.

For a vertex at coordinates (i, j, k), j is considered to be the base that is aligned directly with model position i, and as such is called the primary base of the vertex. k is called the vertex's secondary base, as it is aligned indirectly with model position i^* through position i. The primary bases of the vertices along an alignment path are said to comprise the primary strand of an alignment, while the secondary bases of such vertices form the alignment's secondary strand.

The pattern of edges emanating from a single vertex in this three-dimensional alignment graph is shown in figure 2. Note that when viewed in a direction parallel to the graph's z axis (fig. 2a), the edge pattern is quite similar to the edge pattern used in the two-dimensional alignment graph. The interpretation of these edges is also similar: the projection of an edge onto the xy plane represents the gap occurring in the primary strand of an alignment. The z axis projections of the alignment graph edges (figs. 2b - c) represent gaps in the secondary strand of an alignment. Each edge receives a weight that is the sum of the primary and secondary strand gap penalties.

An important feature of the edge pattern that is exhibited in figures 2b and c is that edges extend from each vertex in both the positive and negative z direction. It is this fact that allows *Seq7* to align sequences to pseudoknotted structures, as illustrated in figure 3. As can be seen there, the z coordinate of the alignment path (representing the alignment's secondary strand) generally decreases as one proceeds in the 5′ to 3′ direction of the alignment. This behavior is allowed by the edges that extend in the $-z$ direction. However, in some places where a boundary between the two helices constituting the pseudoknotted structure is crossed (such as between alignment positions 11 and 12, or across the single-stranded region between positions 3 and 6), the alignment path must move to vertices with a higher z coordinate, necessitating the traversal of edges that extend in the $+z$ direction.

***Seq7*'s shortest path algorithm.** *Seq7* actually uses a slightly modified version of Dijkstra's algorithm to perform structure-based alignments. The first modification is that *Seq7* constructs the edges of the alignment graph "on the fly," as they are needed by the algorithm. This helps to speed up the program, as edges that are never needed do not get constructed. However, this modification was really made because it facilitates the implementation of the second modification, which is needed to ensure that the primary and secondary strands of an alignment remain consistent with each other. As can be seen in figure 2a, the xy projections of the edges are all directed from upper

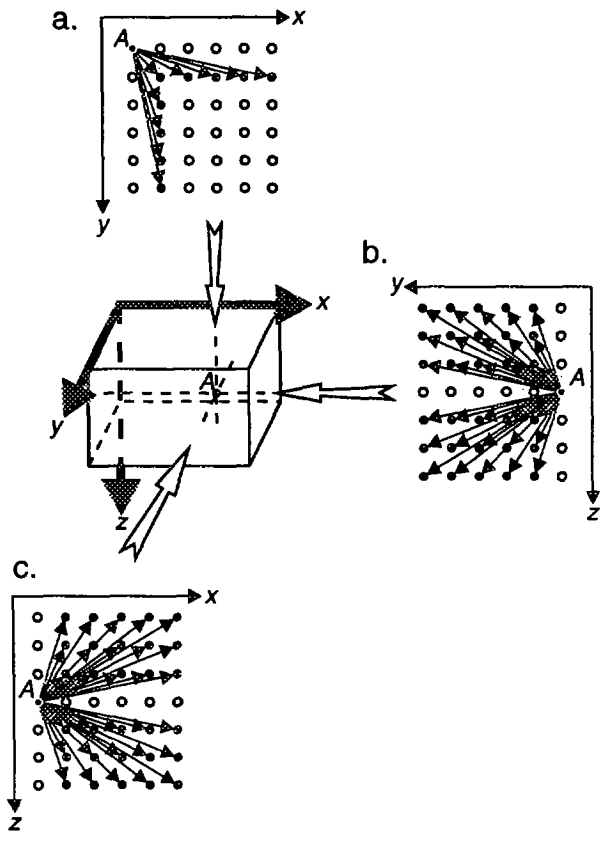
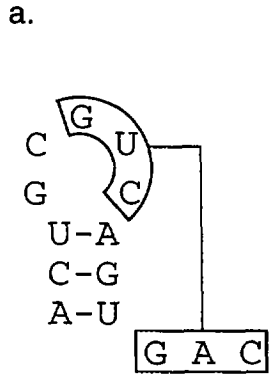


Figure 2: Pattern of edges emanating from a single vertex (labeled A) in the *Seq7* alignment graph. (a) View along the z axis. (b) View along the x axis. (c) View along the y axis.

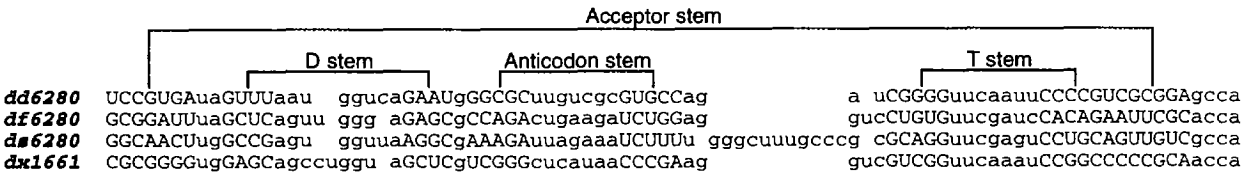


b.

Model Posn.	1° Base	2° Base
1	A1	U11
2	C2	G10
3	U3	A9
4	G4	-
5	C5	-
6	G6	C14
7	U7	A13
8	C8	G12
9	A9	U3
10	G10	C2
11	U11	A1
12	G12	C8
13	A13	U7
14	C14	G6

Figure 3: (a) A pseudoknotted RNA structure, and (b) its representation as the primary and secondary strands of a Seq7 alignment. See text for details.

a. True alignment



b. *Seq7*

```
tRNA    >>>1>>>++>2>>+++ +++++++<<2<+>>3>>+++++++++<3<<++          +++>>4>>+++++++++<<4<<<<<1<<<++++

dd6280  UCCGUGAuaGUUUaau ggucaGAAUgGGCGCuugucgcGUGCCag          au CGGGGuucaauuCCCCGUCGCGGAgcca
df6280  GCGGAUUuaGCUCagu uggg aGAGCgCCAGAcugaagaUCUGGag          gucCUGUGuucgaucCACAGAAUUCGCacca
ds6280  GGCAACUugGCCGagu gguuaAGGCgAAAGAuuagaaaUCUUUugggcuuugcccg cGCAGGuucgaguCCUGCAGUUGUCgcca
dx1661  CGCGGGGugGAGCagccuggu aGCUCgUCGGGcucauaaCCCGAag          gucGUCGGuucaaauCCGGCCCCCGCAacca
```

Figure 4: Alignments of four tRNA sequences of known crystal structure. (a) True alignment. (b) *Seq7*'s alignment. Note that in this and in all alignment figures that follow, bases believed to participate in pairing interactions are capitalized. Also, a representation of the model structure used by *Seq7* appears in the top line of the program's alignments. See figure 5 of Eddy & Durbin (1994) for comparison with other alignment programs.

right to lower left. As such, it is possible to determine which bases of the sequence have already been used in the primary strand of a given alignment from the y coordinate of the final vertex in that alignment path. However, the fact that paths may extend in either z direction means that the bases used in an alignment's secondary strand cannot be determined from the z coordinate of the last vertex on the alignment path. To prevent conflicts, then, before *Seq7* decides how to extend a given alignment path, it traces back along that path to find out which bases are still available to be aligned to the model. The program will then only construct edges that lead to vertices that would not conflict with any previously assigned bases along that path.

Note that these modifications concern only the construction of the alignment graph, and do not affect the shortest path determination process itself.

N best mode Since the "best" alignments reported by *Seq7* are necessarily suboptimal (see below), users may want to know whether alternate alignments exist for a particular sequence and model structure. For this reason, we have implemented an N best mode in *Seq7*.

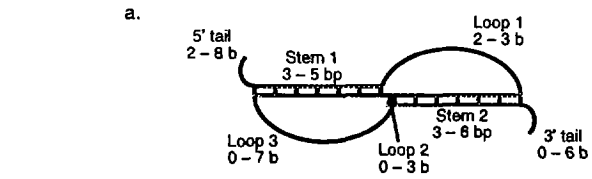
In normal operation, *Seq7* exits immediately after finding the shortest path from the *Start* to the *Stop* vertex of the alignment graph. In N best mode, however, *Seq7* continues building paths through the alignment graph until it has found the shortest path from *Start* to every other vertex in the graph, thereby forming the shortest path spanning tree of the alignment graph. Once this tree has been constructed, the program compiles a list of "complete" alignment paths — those in which every model position and sequence base is accounted for — and sorts the list by path length. The best N alignments (where N is specified by the user) are then reported.

Optionally, users may specify a minimum distance parameter, D. When a D value is given, *Seq7* will not report any alignment that differs by fewer than D positions from any other alignment in the N best list.

***Seq7* performance.** Since *Seq7* creates edges on the fly, and then discards them when they are no longer needed, the program's memory usage is dominated by the storage of the vertices of the alignment graph. So, when aligning a sequence of N bases to a model structure containing M positions, *Seq7*'s memory complexity is $O(MN^2)$. *Seq7*'s time performance in normal mode is difficult to determine analytically, since the program does not construct all of the edges of the alignment graph, but has been found empirically to average $O(M^{1.5}N^{3.1})$. In N best mode, *Seq7* runs in roughly $O(M^2N^4)$ time.

For reference, when running on a Sun SPARCstation 10, *Seq7* requires approximately 4.7 Mb of memory and 18.4 cpu seconds to perform the tRNA alignment depicted in figure 4.

As stated in the introduction, the problem of aligning a sequence to a nonplanar structure allowing for variable-length gaps is known to be NP-hard (Lathrop 1994), so optimality had to be sacrificed to achieve computational efficiency. *Seq7*'s suboptimality arises in the fact that the program must occasionally choose



b.

```
HIV_RT  +++ +++>> 1>> ++>>2>>>+ + <<1<<+ +++  +<<<2<<  + +++

1_1      a    aUU CCG uuUUCAGU    CGGGAa  aa   ACUGAAcaa ucu
1_3a    aau aucUU CCG aaGCCG  a a CGGGAa  aa  c  CGGC  a ucu
1_7    gaau cga C CC  aaGCC   aaa  GGG gauaaugc   GGC
1_17    uaa ugA C CAG agGCCC  a a CUGG   Uaa ac  GGGC  ggucu
2_4a     uagaua U CCG aaGCUCA a   CGGG a uaa    UGAGC  a ucu
2_7a    aua uga U CCGuaaGAGG  a   CGGG a uaa  a  CCUC  a acu
2_9          guUU CCG aaAGA   a auCGGGAa  aacug   UCU
2_11     uc auauuaCCGuuaCUCCU     CGGga  uaa    AGGAG  a ucu
```

Figure 5: Alignment of HIV RT inhibitor RNAs. (a) Proposed structure (Tuerk et al. 1992). (b) *Seq7* alignment.

a secondary base for an alignment position without accounting for any gaps that will need to be created to accommodate that assignment. However, it has been found in practice that the model structure usually contains sufficient information for *Seq7* to choose these bases correctly. There is no guarantee, though, that the program will always do so.

Automated alignment construction

Recently, we have developed a graph-theoretical RNA structure prediction method which is able to find the optimal nonplanar structure of an alignment of RNA sequences in polynomial time (Cary & Stormo 1995; Tabaska et al. 1997). This method is implemented in a program called *wmatch*. *wmatch* may be used in conjunction with *Seq7* to form an Expectation-Maximization (EM) method for automating the process of generating RNA sequence alignments. This procedure is essentially as described by Eddy and Durbin (1994), except that through the use of *Seq7* and *wmatch*, we are able to generate alignments that can reflect the presence of pseudoknots.

There are two ways in which *Seq7* and *wmatch* can be used to generate alignments from scratch. The first begins with the user supplying a seed alignment to start the EM alignment process. This alignment may be constructed so as to reflect any structural or functional information that is known about the RNAs of interest. If such information is not available, though, one could simply align the sequences by their 5′ ends. In the E step of the EM algorithm, *wmatch* is used to predict the structure of the RNAs based on the seed alignment. Then, in the M step, *Seq7* realigns the sequences to *wmatch*'s predicted structure. Cycles of structural modeling and realignment continue until the alignment converges.

The second way to generate alignments using *wmatch* and *Seq7* is similar to the procedure outlined above, except that the process starts with a seed structure. This approach can be useful if one has a partial structure for the RNAs of interest, or knows the structure of a related RNA. It is also possible to incorporate functional information in a seed structure through a technique we call "softening," in which the local insertion, deletion, and base mismatch penalties of those model positions outside of the functionally important regions of the molecule are reduced. This softening makes it relatively more expensive for *Seq7* to misalign the known functional elements of a set of RNAs, resulting in a better alignment.

Availability

Seq7, *wmatch*, and related support programs are available online. Send E-mail requests to: jtabaska@ural.colorado.edu

Results and discussion

Seq7

In figure 4, *Seq7* was used to align the sequences of four tRNAs of known structure to a generic tRNA model. As can be seen, *Seq7* correctly identifies all of the bases constituting the four main stems of the structure, and handles the variable loop of sequence ds6280 properly, as well. There are three misplaced bases in the single-stranded regions of the various sequences, but each of these errors is in the direction of increased sequence conservation; the placement of these bases in the true alignment is based on structural or functional information which was not made available to *Seq7* in the tRNA model structure.

Figure 5 illustrates *Seq7*'s ability to align RNA sequences to pseudoknotted structures. The RNAs used

a. Published alignment


```
a0660   aGCTCagtt ggg  aGAGCgCTGCCcttgcaaGGCAGag
d1660    GTTCagtc ggtt aGAATaCCTGCctgtcacGCAGGggg
e6320  tgGTCCaac  ggct aGGATtCGTCGctttcacCGACGcgg
i1180   aGCTCagtt ggtt aGAGCaTCCGGctcataaCCGGAtggt
l1940  tgGCGGaatt ggta gACGCgCATGGttcaggtCCATGtg
l6280    GCCGagt  ggtctaAGGCgCCAGActcaagtTCTGGtctt
p7560  tgGTCTagg  ggt  aTGATtCTCGCttcgggtGCGAGaggt
p8100  tgGTCTagg  ggt  aTGATtCTCGCttcgggtGCGAGagg
s1542    GCCGagc  ggttgaAGGCaCCGGTcttgaaaACCGGcga
s8040   gGCCGagt  ggtt aAGGCgATGGActagaaaTCCATtggg
v2840  taACTCagc  ggt  aGAGTgTCACCttgacgtGGTGGaagt
w6160   gGCGCaac  ggt  aGCGCgTCTGActccagaTCAGAag
x9990  taGTTCagactggt  aGAACgGCGGActgtagaTCCGCatg
y2520   gCCCGagc  ggttaaTGGGgACGGActgtaaaTTCGTtggc
y3200    GCCAagtt ggtttaAGGCgCAAGActgtaaaTCTTGaga
```

b. Seed alignment

```
a0660   aGCTCagttgggaGAGCgCTGCCcttgcaaGGCAGag
d1660   GTTCagtcggttaGAATaCCTGCctgtcacGCAGGggg
e6320   tgGTCCaacggctaGGATtCGTCGctttcacCGACGcgg
i1180   aGCTCagttggttaGAGCaTCCGGctcataaCCGGAtggt
l1940   tgGCGGaattggtagACGCgCATGGttcaggtCCATGtg
l6280   GCCGagtggtctaAGGCgCCAGActcaagtTCTGGtctt
p7560   tgGTCTaggggtaTGATtCTCGCttcgggtGCGAGaggt
p8100   tgGTCTaggggtaTGATtCTCGCttcgggtGCGAGagg
s1542   GCCGagcggttgaAGGCaCCGGTcttgaaaACCGGcga
s8040   gGCCGagtggttaAGGCgATGGActagaaaTCCATtggg
v2840   taACTCagcggtaGAGTgTCACCttgacgtGGTGGaagt
w6160   gGCGCaacggtaGCGCgTCTGActccagaTCAGAag
x9990   taGTTCagactggtaGAACgGCGGActgtagaTCCGCatg
y2520   gCCCGagcggttaaTGGGgACGGActgtaaaTTCGTtggc
y3200   GCCAagttggtttaAGGCgCAAGActgtaaaTCTTGaga
```

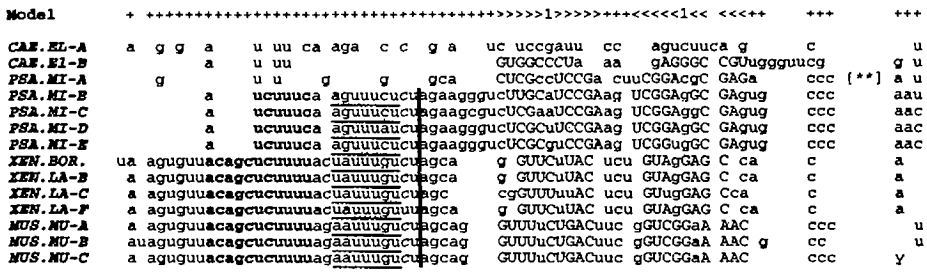
c. Generated alignment

```
Model    +>1>> +++++++ + + <<1<>>2>>>++++++++<<<2< <++

a0660    aGCTC agttggg   a GAGCgCTGCCcttgcaaGGCAG a g
d1660     GTTC agtcggt t a GAATaCCTGCctgtcacGCAGG ggg
e6320   tgGTCC aa cggc t a GGATtCGTCGctttcacCGACGcg g
i1180    aGCTC agttggt t a GAGCaTCCGGctcataaCCGGA tgg t
l1940   tgGCGG aattggt a g ACGCgCATGGttcaggtCCATG t g
l6280     GCCG agt ggtct a AGGCgCCAGActcaagtTCTGG tct t
p7560    tgGTCTag  ggg t aTGATt CTCGCttcgggtGCGAG  aggt
p8100    tgGTCTag  ggg t aTGATt CTCGCttcgggtGCGAG  agg
s1542     GCCG ag cggt tga AGGCaCCGGTcttgaaaACCGG cga
s8040    gGCCG agt ggt t a AGGCgATGGActagaaaTCCAT tggg
v2840   taACTC ag cggt   a GAGTgTCACCttgacgtGGTGG aag t
w6160    gGCGC aa cggt   a GCGCgTCTGActccagaTCAGA a g
x9990   taGTTC agactgg t a GAACgGCGGActgtagaTCCGCat g
y2520    gCCCG ag cggt t aaTGGGgACGGActgtaaaTTCGT tgg c
y3200     GCCA agttggt tta AGGCgCAAGActgtaaaTCTTG aga
```

Figure 6: Automatically generated alignment of tRNA sequence fragments. (a) Known alignment of the sequences. (b) Seed alignment. (c) Resulting alignment.

a.Generated alignment



b. Published alignment

Figure 7: (a) Automatically generated alignment of U7 snRNA sequences, starting from a seed structure containing functional information. [**] marks where a 19-base segment (UUCUAGAGAAACUUGAAAG) was removed from sequence PSA.MI-A because of space limitations. (b) Alignment of the same sequences from the uRNA database (Zwieb 1996). Functional elements are marked as follows: Histone pre-mRNA pairing region in boldface; Sm antigen-binding region underlined; histone pre-mRNA cleavage site with a vertical line.

in this figure are a set of *in vitro* selection products that bind and inhibit HIV reverse transcriptase (HIV RT) (Tuerk et al. 1992). The sequences were aligned to a consensus model structure based on the structures predicted by Tuerk et al. (1992) (fig. 5a). As can be seen in figure 5b, *Seq7*'s alignment of the HIV RT inhibitors reflects the supposed structure of the molecules very well, even in the highly variable Helix 2 region.

Automated alignment

Figure 6 shows an alignment generated by our *Seq7/wmatch* EM procedure. The sequences used (fig. 6a) were fragments of 15 sequences randomly selected from a database of tRNA genes (Steinberg et al. 1993). The fragments all included the D and Anticodon stems of the tRNAs. Some fragments also contained extra bases from the flanking single-stranded regions, to make it somewhat more difficult for the procedure to find the correct alignment. To further increase the difficulty, the seed for the alignment process (fig. 6b) was constructed simply by aligning the sequences by their 5′ ends. The EM alignment procedure was able to find the two helices in the tRNA fragments and generate an alignment that very accurately reflects this structure (fig. 6c). The automatically generated alignment also correctly positions the conserved functional elements in the sequences, including the highly variable anticodons. It is interesting to note that the most obvious "error" in the alignment of figure 6c, the slightly misaligned D stems of sequences p7560 and p8100, were actually created to avoid generation of U-U base pairs in these sequences.

Figure 7 depicts an alignment of U7 snRNA sequences generated from a seed structure. The seed used here was based on the proposed structure of *Xenopus spp.* U7 snRNA (Phillips et al. 1992). This seed structure also contained functional information in the form of softened nonfunctional positions. In this case, model positions outside of the histone pre-mRNA base-pairing region, the Sm antigen binding site, the pre-mRNA cleavage region, and the hairpin were softened by lowering their positional insertion, deletion, and mismatch penalties. The resulting alignment (fig. 7a) compares favorably with the manually-constructed alignment (fig. 7b) obtained from the uRNA database (Zwieb 1996) in the hairpin region of the molecule. However, the functional information incorporated into the seed structure allowed the EM alignment procedure to construct an alignment which more accurately reflects the presence of the known functional elements than does the manual alignment.

Conclusion

We have developed a graph theoretical method for performing multiple structure-based RNA sequence alignments. Our method, implemented in the program *Seq7*, is able to align RNA sequences to pseudoknotted structures allowing for variably-sized gaps, while paying a small penalty in optimality. We have also shown that *Seq7* can be used in combination with our RNA-folding program *wmatch* to discover the structure of RNAs and perform alignments based on that structure. As such, *Seq7* and *wmatch* should be useful tools for RNA researchers.

Some readers may be concerned about the degree of *Seq7*'s suboptimality. *Seq7* may produce an incorrect alignment when there exist several ways to form a particular helix in the model structure with a given sequence. This really only occurs when the model structure contains short (1 – 3 bp) helices of ill-defined sequence. Furthermore, *Seq7* assesses penalties that help maintain proper spacing of structural elements, so suboptimal alignments are generally only produced when there are several overlapping ways to form a helix, which occurs very infrequently. Nevertheless, it can happen, which is why we have implemented of *Seq7*'s N best mode.

Consideration of the alignment graph edge pattern shown in figure 2 reveals that *Seq7* should be able to align sequences to parallel helices. In fact, in the current version of the program, a heuristic has been implemented to prevent this from happening. There is, however, no reason why this rule couldn't be relaxed so that later versions will be able to generate alignments to these unusual tertiary structures. Another intriguing possibility is that the alignment graph can be expanded to four dimensions in some regions, allowing for alignments to base triples. This could be a useful capability since we have developed a variant on our *wmatch* program, called *bmatch*, which is able to detect base triples in RNAs, along with other types of tertiary base interactions (Tabaska et al. 1997). Therefore, through the use of an expanded *Seq7* and *bmatch* in an EM algorithm as described above, we may soon be able to automate the detection of many new and unusual RNA structural elements from raw sequence data.

Acknowledgements

This work was supported primarily by DOE grant ER61606, and partially by NIH grant HG00249.

References

Cary, R. B. and Stormo, G. D. 1995. Graph-Theoretic Approach to RNA Modeling Using Comparative Data. In Proceedings of the Third International Conference on Intelligent Systems in Molecular Biology 75 – 80. Menlo Park, Calif.: AAAI Press.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1: 269 – 271.

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

Gautheret, D.; Major, F.; and Cedergren, R. 1990. Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. *CABIOS* 6: 325 – 331.

Kim, J.; Cole, J. R.; and Pramanik, S. 1996. Alignment of possible secondary structures in multiple RNA sequences using simulated annealing. *CABIOS* 12: 259 – 267.

Lathrop, R. H. 1994. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Engineering* 7: 1059 – 1068.

Phillips, S. C. and Binrstiel, M. L. 1992. Analysis of a gene cluster coding for the *Xenopus laevis* U7 snRNA. *Biochimica et Biophysica Acta* 1131: 95 – 98.

Sakakibara, Y.; Brown, M.; Mian, I. S.; Underwood, R.; and Haussler, D. 1994. Stochastic context-free grammars for modeling RNA. In Proceedings of the Hawaii International Conference on System Sciences 284 – 293. Los Alamitos, Calif.: IEEE Computer Society Press.

Steinberg, S.; Misch, A.; and Sprinzl, M. 1993. Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Research* 21: 3011 – 3015.

Tabaska, J. E.; Cary, R. B.; Gabow, H. N.; and Stormo, G. D. 1997. An automated RNA modeling approach capable of identifying pseudoknots and base-triples. Forthcoming.

Tuerk, C.; MacDougal, S.; and Gold, L. 1992. RNA pseudoknots that inhibit human immunodeficiency virus type 1 reverse transcriptase. *Proceedings of the National Academy of Sciences USA* 89: 6988 – 6992.

Zwieb, C. 1996. The uRNA database. *Nucleic Acids Research* 24: 76 – 79.