

Universität Bielefeld

Technische Fakultät
Abteilung Informationstechnik
Forschungsberichte

From RNA Folding to Thermodynamic Matching, including Pseudoknots

Robert Giegerich

Jens Reeder

Report 2003-03



Impressum: Herausgeber:
Robert Giegerich, Ralf Hofestädt, Franz Kummert, Peter Ladkin,
Helge Ritter, Gerhard Sagerer, Jens Stoye, Ipke Wachsmuth

Technische Fakultät der Universität Bielefeld,
Abteilung Informationstechnik, Postfach 10 01 31,
33501 Bielefeld, Germany

ISSN 0946-7831

From RNA Folding to Thermodynamic Matching, including Pseudoknots

Robert Giegerich, Jens Reeder

Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany

ABSTRACT

Motivation: The general problem of RNA secondary structure prediction under the widely used thermodynamic model is known to be NP-complete when the structures considered contain arbitrary pseudoknots. For restricted classes of pseudoknots, several polynomial time algorithms have been designed, where the $O(n^6)$ time and $O(n^4)$ space algorithm by Rivas and Eddy is currently the best available program.

Results: We introduce the class of canonical simple recursive pseudoknots and present an algorithm that requires $O(n^4)$ time and $O(n^2)$ space to predict the energetically optimal structure of an RNA sequence, possible containing such pseudoknots.

The new algorithm is presented in rather abstract form, such that, mainly by copy-and-paste, thermodynamics based matchers can be derived from it. Such matchers recognize the energetically best structure that fits a predefined motif. This approach is demonstrated by developing a matcher for the consensus structure of group I introns, which contains a pseudoknot of the class considered here.

Availability The algorithms *pknotsRG*, *bestknot*, and *glintron* are available from the first author upon request, and will also be installed on the Bielefeld Bioinformatics Server[†].

Contact: robert@techfak.uni-bielefeld.de

Keywords: RNA pseudoknots, structural motifs

INTRODUCTION: THE PSEUDOKNOT CHALLENGE

Biological relevance Pseudoknots have been shown to be functionally relevant in many RNA mediated processes. Examples are the self-splicing group I introns [Cech (1988)], ribosomal RNAs, or RNaseP. Recently, pseudoknots were located in prion proteins of humans, and confirmed for many other species [Barette et al. (2001)]. With the current increased interest in the universe of RNA functions [Dennis (2002)], algorithmic support for analysing structures that include pseudoknots is much in demand.

Previous algorithmic work Well established algorithms for the prediction of RNA secondary structures (MFOLD, [Zuker and Sankoff (1984)], RNAfold [Hofacker et al. (1994)]) are commonly based on a thermodynamic model [Turner et al. (1988)], returning a structure of minimal free energy, called MFE-structure for short. In spite of their importance, pseudoknots are excluded from consideration by these programs for reasons of computational complexity: While folding a sequence of length n into unknotted structures requires $O(n^3)$ time and $O(n^2)$ space, finding the best structure including arbitrary pseudoknots has been proved to be NP-complete [Akutsu (2000); Lyngsø and Pedersen (2001)]. In fact, the proof given in [Lyngsø and Pedersen (2001)] uses a scoring scheme based on adjacent base pairs only, simpler than the MFE model because it neglects entropic energies from loops. These complexity results leave two routes to achieve practical algorithms.

The first route is to consider pseudoknots in full generality, but resort to an even more simplistic energy model. An $O(n^4)$ time and $O(n^3)$ space algorithm for base pair maximization has been given in [Akutsu (2000)], and an $O(n^3)$ time algorithm based on maximum weight matching in [Tabaska et al. (1998)].

The second route is the one followed here: We retain the established thermodynamic model, but restrict to a more tractable subclass of pseudoknots.

[†] <http://bibiserv.techfak.uni-bielefeld.de/>

For some quite general classes of pseudoknots, polynomial time algorithms have been designed: Rivas and Eddy achieve $O(n^6)$ time and $O(n^4)$ space [Rivas and Eddy (1999)]. This algorithm is available, and, in spite of the high computational cost, it is actually used in practice. We shall call it *pknotsRE* for later reference. Further improvements have been shown to be possible for yet more restricted classes, e.g. the non-recursive simple pseudoknots considered by Lyngsø and Pedersen [Lyngsø and Pedersen (2000)] with $O(n^5)$ time and $O(n^4)$ space, but to our knowledge, no implementations are available.

Our contributions The new contributions reported here are the following:

- We present an algorithm *pknotsRG* for folding RNA secondary structures including pseudoknots under the MFE model which requires $O(n^4)$ time and $O(n^2)$ space.
- The algorithm considers the class of simple recursive pseudoknots, further restricted by three rules of canonization. Each simple recursive pseudoknot has a canonical representative that is recognized by *pknotsRG*.
- While this class is more restricted than the one of the Rivas/Eddy algorithm, practical evaluation shows that our algorithm finds the same pseudoknots in all cases tested, while the length range of tractable sequences is increased significantly.
- *pknotsRG* is described and implemented on a high level of abstraction and can therefore serve as a template for MFE-based matchers for specific motifs involving pseudoknots. This is demonstrated for the case of group-I-introns.
- Our first experience shows that large pseudoknots determined by comparative studies may be rather far from MFE-structures, both in terms of shape and energy. This implies that we cannot hope to detect large pseudoknots by thermodynamic folding, but only by use of thermodynamic matchers as described here.

METHODS AND MODELS: SIMPLE AND CANONIZED RECURSIVE PSEUDOKNOTS

It is not easy to relate the classes of pseudoknots recognized by the different algorithms mentioned above. We refer the reader to the review by Lyngsø and Pedersen [Lyngsø and Pedersen (2001)], which compares these classes by means of examples. The starting point of our work is the algorithm *pknotsRE* by Rivas and Eddy. It recognizes pseudoknots

that can be nested and can have unlimited chains of helices involved in crosswise interactions. The drawback of this powerful, but computationally expensive algorithm is the following paradox: Pseudoknots with complex helix interactions naturally require longer primary sequence than simpler ones. The high runtime complexity of $O(n^6)$, however, as well as the space consumption of $O(n^4)$ restricts the use of this algorithm to a maximal sequence length of around 130–140 nucleotides. Most of the pseudoknots predicted belong to a much simpler structural class and do not exhibit chains of crosswise interactions.

The algorithm developed here achieves time complexity $O(n^4)$ and space complexity $O(n^2)$. The runtime improvement, compared to *pknotsRE*, results from an idea of canonization, while the space improvement results from disallowing chained pseudoknots. These improvements extend the range of tractable sequences to a length up to 400 nucleotides, and we can locate pseudoknots up to this size in even longer sequences. Yet, in a new guise, the paradox persists: We can now find larger pseudoknots, including nested ones, but not those with chained helix interactions that would be found by *pknotsRE*.

Simple recursive pseudoknots

Following the terminology of [Akutsu (2000)], a *simple* pseudoknot crosswise interaction of two helices, as shown in Figure 1. In simple *recursive* pseudoknots, we allow the unpaired strands u, v, w in a simple pseudoknot to fold internally in an arbitrary way, including simple recursive pseudoknots. Let us call this class sr-PK. More complex knotted structures like triple crossing helices or kissing hairpins, as shown in Figure 3, are excluded from sr-PK. We will show later how they can be integrated in our approach and outline the increased computational cost of doing so. For the main part of this paper, we concentrate on the class sr-PK.

Anticipating the complexity of a DP algorithm

Thermodynamic RNA folding is implemented via dynamic programming (DP). We start with a semi-formal discussion of how to estimate the efficiency of a DP algorithm for motif search *before* it is written in detail. We consider elements of RNA structure as sequence motifs of different types: hairpins, bulges, multiloops, etc. By an equation

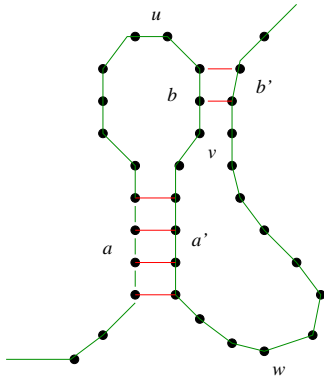


Fig. 1. A simple pseudoknot, formed by helices $a - a'$ and $b - b'$, with intervening sequences u, v, w .

$$\begin{array}{l}
 m = f \lll a \sim\sim\sim b \sim\sim\sim c \lll \\
 g \lll c \sim\sim\sim a
 \end{array}$$

we specify that the sequence motif m can be composed in two alternative ways: The first case, labelled by f , requires adjacent occurrences of motifs a, b , and c . The second case, labelled by g , requires adjacent occurrences of motifs c and a . When motif m is to be scored, f and g are seen as the scoring functions that combine the local score contribution of each case with the scores of sub-motifs a, b , and c .

What is the computational effort of locating motif m in an input sequence x of length n , say at sequence positions i through j ? First we assume that all motifs can have arbitrary size between 0 and n . The algorithm must consider all boundary positions (i, j) for motif m , which requires $O(n^2)$ steps at least. In case g , it must consider all boundary positions k where motifs c meets a , such that the runtime for case g is in $O(n^3)$. In case f , there are two such moving boundaries k and l between the three sub-motifs, so we obtain $O(n^4)$ overall for motif m .

This can be improved if there is an upper bound on the size of some motif involved. If motif a is a single base, for example, the exponent of n decreases by 1 in both cases. Furthermore, if motif b is (say) a loop of maximal size 40, then one factor of n is reduced to a constant factor and overall asymptotic runtime is now $O(n^2)$. Sometimes a motif description can be restructured to improve efficiency by reducing the number of moving boundaries. Whether or not this is possible does not depend on the motif structure, but on the scoring scheme. Such optimizations are studied in

[Giegerich and Meyer (2002)], where also the line of reasoning exercised here is given a mathematical basis.

In the sequel, we shall exploit another source of efficiency improvement. If the lengths of two sub-motifs are coupled somehow, say a and c have the same length, then the boundaries k and l in case f are related by $k - i = j - l$. When iterating over k , we can use $l := j - k + i$ (rather than $k \leq l \leq j$) and save another factor of n .

Canonization

When the search space of a combinatorial problem seems to be too complex to be evaluated efficiently, heuristics are employed. Canonization restricts the search space in a well-defined way, arguing that all the relevant solutions in the full search space have a representative that is canonical, and hence, nothing relevant is overlooked. One such technique is the purging of structures that have isolated basepairs. Here the plausibility argument refers to the underlying energy model, where base pairings without stacking have little or no stabilizing effect. This canonization does not affect efficiency, but it achieves a significant reduction of the search space (figures in Giegerich (2000a)), which renders the enumeration of near-optimal solutions [Wuchty et al. (1998)] much more meaningful.

We shall introduce three canonization rules that reduce class sr-PK to the class of *canonized simple recursive pseudoknots*, csr-PK.

Using the notation introduced above, the motif definition of a simple recursive pseudoknot is given by

$$\text{knot} = \text{knt} \lll a \sim\sim\sim u \sim\sim\sim b \sim\sim\sim v \sim\sim\sim a' \sim\sim\sim w \sim\sim\sim b'$$

with boundaries at sequence positions i, e, k, g, f, l, h, j as shown in Figure 2. Segment a forms a helix with a' , and b with b' . Segments u, v , and w can have arbitrary structures, including pseudoknots. Naively implemented, we can expect a DP algorithm of time complexity $O(n^8)$ according to our efficiency estimation technique introduced above. We now apply canonization. Note that it only applies to helices forming pseudoknots; other helices are unaffected. We first present the technical aspects; the discussion of these restrictions is deferred to the next section.

Canonization Rule 1 (a) Both strands in a helix must have the same length, i.e. $|a| = |a'|$ and $|b| = |b'|$. (b) Both helices must not have bulges.

Note that (b) is a stronger restriction and trivially implies (a). Under the regime of Rule 1 we may

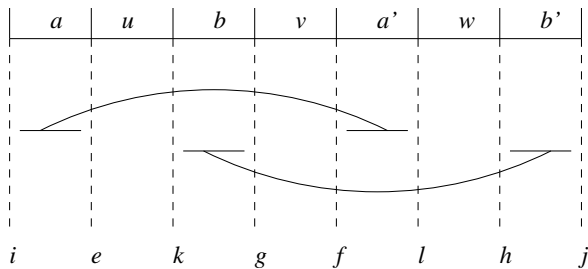


Fig. 2. Eight moving boundaries delineating a simple recursive pseudoknot

conclude:

$$\begin{aligned} f &= l - (e - i) \\ h &= j - (g - k) \end{aligned}$$

We are left with 6 out of 8 boundaries that vary independently, and runtime is down to $O(n^6)$.

Canonization Rule 2 The helices a, a' and b, b' facing each other must have maximal extent, or in other words, compartment v must be as short as possible under the rules of base pairing.

We observe that the maximal length of a and a' is fixed once i and l are chosen. The maximal helix length $stacklen(i, l)$ can be precomputed and stored in an $O(n^2)$ table. The same observation holds with respect to the other helix, and we fix

$$\begin{aligned} e &= i + stacklen(i, l) \\ g &= k + stacklen(k, j). \end{aligned}$$

Thus, we are left with only four independently moving boundaries – i, k, l, j –, and can hope to obtain an algorithm with runtime $O(n^4)$. Scores of pseudoknots found between i and j are stored in table $knot(i, j)$, and hence the space requirements are $O(n^2)$, which is the same asymptotic efficiency class as in the folding of unknotted structures.

A subtlety arises when both helices, chosen maximally, compete for the same bases of v , or in other words, the length of v would become negative. This case is addressed by

Canonization Rule 3 If two maximal helices would overlap, their boundary is fixed at an arbitrary point between them.

Let m and m' be the helix lengths so determined. We finally obtain

$$\begin{aligned} e &= i + m \\ g &= k + m' \end{aligned}$$

The language of pseudoknots in class $csr\text{-}PK$ can be defined by a simple context free grammar over an infinite terminal alphabet. Let a^k denote a terminal symbol of k times the letter a . The grammar uses a single nonterminal symbol S and its productions are

$$S \rightarrow . \mid .S \mid S. \mid SS \mid (S) \mid [^k S \{^l S \}^k S]^l$$

for arbitrary $k, l \geq 1$. For example, the simple pseudoknot of Figure 1 is represented as the string $..[[[[[.....\{\{..\}]]]].....\{\}]]$.

This grammar is useful to judge how far an experimentally determined structure is from class $csr\text{-}PK$. It is not useful for programming, since it is ambiguous and does not distinguish the fine grained level of detail required in the energy model. The implemented algorithm corresponds to a non-ambiguous grammar using 22 nonterminal symbols and 56 productions.

Canonical representatives

A careful discussion is required to show that each simple recursive pseudoknot, if not canonical by itself, has (a) a canonical representative of (b) similar free energy.

Rule 1 (b) affects the length of helices that are considered in forming the pseudoknot. Let there be a pseudoknot between i' and j' . It is not canonical if one of the two helices contains bulges. However, there must be at least one pair of shorter helices without bulges at i, j with $i' \leq i$ and $j \geq j'$, which serves as a canonical representative, albeit with somewhat higher free energy.

Rule 2 is justified by the fact that the energy model strongly favors helix extension. Clearly, for each family of pseudoknots delineated by i, k, l, j there is a canonical one with maximal helices, whose free energy is at least as low – except for the following case: The maximal helices compete with the internal structure of u, v and w . It may be possible to contrive a structure where shortening (say) helix (a, a') by one base pair to create two pairs with new partner bases in u and v , resulting in a structure which has slightly lower energy. Still, the free energy of the canonical pseudoknot should be very similar.

Finally, Rule 3 requires a decision where to draw the border between two helices facing each other and competing for the same bases. An arbitrary decision here can only slightly affect free energy, as

RNA function	N	within csr-PK	1 NT bulge	Rule violated	not in sr-PK
viral frame-shifting	20	16	3	0	(kiss) 1
ribozymes	9	3	2	(CR 2) 1	(kiss) 2 (4 helices) 1
mRNA	11	4	0	(CR 1) 4	(3 helices) 1

Table 1. Class membership of 40 pseudoknots from PseudoBase that were determined by comparative sequence analysis and/or by experimental techniques.

the same base pairs are stacked either on the $a - a'$ or the $b - b'$ helix.

Let $E(s)$ denote the free energy computed for structure s . Summing up, we have shown that for each simple recursive pseudoknot K , there is a canonical one C in the search space. While we cannot prove that $E(C) \leq E(K)$, we have argued that this is likely, and if not, the energies will at least be close. Still, there might be another, energetically optimal canonical structure S (knotted or not) such that $E(K) < E(S) < E(C)$. In this case, if only the “best” structure S is reported, neither K nor its canonical representative C is observed.

Pragmatics

To evaluate empirically the class csr-PK, we considered 40 pseudoknot structures from PseudoBase[‡]. The observations are shown in Table 1. We find 35 simple recursive pseudoknots, and 5 of more general shapes. We find that 23 out of the 35 pseudoknots lie in csr-PK, i.e. they are their own canonical representatives. 5 more will fall into this class if we allow a 1 nucleotide bulge in Canonization Rule 1, totalling 28 out of 35. Currently, their canonical representatives will have their helices shortened to exclude the bulge.

Note that often pseudoknots in more general classes also have a good representatives in csr-PK. For example, one pseudoknot (HDV-It_g) consists of four interacting helices of shape $a - b - c - d - c' - a' - d' - b'$, where helix $d - d'$ is very short - only two base pairs. Deleting it, helix $c - c'$ is no longer interacting with other helices, and the pseudoknot falls within class csr-PK. By the way, all 5 pseudoknots of the more general shapes satisfy our canonization rules.

To deal pragmatically with the problem when

the optimal (knotted) structure is non-canonical, and its canonical representative is dominated by an unrelated structure, we provide two means: First of all, our algorithm is non-ambiguous, the prerequisite for a non-redundant enumeration of near-optimal structures [Giegerich (2000a)]. We can have the program to report the k best structures. Secondly, we provide an option in our program to compute the best[§] canonical pseudoknot that can be formed (be it within the MFE structure or not) *somewhere* in the input sequence. This is achieved by adding two clauses:

```
bestPK = skipleft <<< base ~~~ bestPK ||| bestPK1
bestPK1 = skipright <<< bestPK1 ~~~ base ||| knot
```

These clauses have time complexity $O(n^2)$ and preserve the non-ambiguity of the algorithm. If desired, an enumeration of near-optimal “best” pseudoknots is also feasible.

IMPLEMENTATION VIA ALGEBRAIC DYNAMIC PROGRAMMING

Using the ideas presented so far, our folding algorithm can be implemented in any language suitable for dynamic programming, say FORTRAN or C. However, this would preclude our second goal, the use of this algorithm as a template for the systematic derivation of thermodynamic matchers. We use a more high-level approach.

A review of the ADP method

The folding algorithm was developed and implemented using the method of algebraic dynamic programming (ADP) [Giegerich and Meyer (2002); Giegerich (2000b)]. In ADP, the search space of a DP problem is defined on a declarative level, specified by clauses like the ones we have already seen above. Together they form a tree grammar, defining a tree language whose elements are all the candidates in the search space. In our case, the candidates are RNA structures represented as trees. The typical DP recurrences are implicit in this description. Scoring is achieved by interpreting the operators (e.g., *knt*, *skipleft*, *skipright*) that build the trees as scoring functions. The grammar needs to be annotated with respect to tabulation and the application of the objective function (e.g., minimization).

The advantage of this method is its high level of abstraction. No subscripts, no errors. The perfect

[‡]<http://wwwbio.LeidenUniv.nl/Batenburg/PKB.html>

[§] “Best” is defined here in minimal free energy per base, to avoid a built-in bias towards large pseudoknots.

separation of search space definition and evaluation allows the same grammar to be used for different kinds of analyses. Furthermore, important algorithmic properties such as non-ambiguity and efficiency can be studied on this level of abstraction. Last not least, an ADP program can be executed as is, avoiding the explicit formulation of DP recurrences (and a whole universe of programming errors). A significant, but constant factor of speedup can be gained by explicitly formulating the recurrences and implementing them in a lower level language. (Automating this process is part of our current work.)

We start from an ADP algorithm for folding RNA secondary structures (excluding pseudoknots) provided by Dirk Evers. We show ADP clauses defining the closed substructures: stacks, hairpins, bulges, and multiloops, adding an alternative for pseudoknots:

```
closed = (stack ||| hairpin ||| leftB ||| rightB |||
         iloop ||| multiloop ||| knot) 'with' basepair
stack = sr <<< base ~~~ closed ~~~ base
hairpin = hl <<< base ~~~ (region 'with' minsize 3) ~~~ base
leftB = bl <<< base ~~~ region ~~~ closed ~~~ base
rightB = br <<< base ~~~ closed ~~~ region ~~~ base
iloop = il <<< base ~~~ inloop ~~~ base
multiloop = ml <<< base ~~~ ml_components ~~~ base
```

The shown code abstracts from efficiency annotation and the treatment of dangling bases. The complete algorithm is found on the ADP pages[†]. It is based on the standard MFE model with coaxial stacking and dangling bases, includes Lyngsø et al.'s improved treatment of internal loops [Lyngsø et al. (1999)], is non-ambiguous and requires $O(n^3)$ time and $O(n^2)$ space. Closed substructures are defined such as to avoid lonely base pairs. While all this is easily expressed within the standard ADP framework, our new algorithm requires extensions which are now explained.

Adding pseudoknots

The implementation strictly follows the outline given in the methods section, except that a considerable amount of detail related to the energy model has to be taken care of. While ADP bans the use of subscripts, our canonization ideas require to explicitly manipulate subscripts. We show the concrete pseudoknot code, but explain only the essential points. A subscript pair (i, j) denotes input sequence positions $inp_{i+1} \dots inp_j$. [...] denotes lists, and \leftarrow denotes choice from a list of alternative values.

```
knot (i,j) = [x | k <- [i+2 .. j-1],
             x <- pknot_int k (i,j)]
pknot_int k (i,j) = [pk' energy a u b v a' w b' l
                    l <- [k+1 .. j-2],
```

These lines choose k and l , and put together the results from a, u, b, v, a', w, b' under the scoring function pk' . The helices $a - a'$ and $b - b'$ should not overlap each other, and not exceed $[k \dots l]$. Furthermore each helix must have a minimum length of two bases. Due to stereochemical reasons one base in the front part and two bases in the back part are left explicitly unpaired; these bases should bridge the stacks. This consideration is taken over from *pknotsRE*. The next four definitions implement canonization rules 1, 2 and 3. They determine the helix lengths, finally computed into the variables m and m' .

```
alphen = min (stacklen (i,l)) (1-k-2),
betalen = min (stacklen (k,j)) (1-k-2),
(m,m') = if (betalen + alphen) > (1-k)
          then divide (k, l, alphen, betalen)
          else (alphen, betalen),
m >= 2, m' >= 2,
```

The next lines define the pseudoknot components a through b' , plus the local energy contribution[‡].

```
a <- region (i, i+m),
u <- front j (i+m+1, k),
b <- region (k, k+m'),
v <- middle (j-m') (i+m) (k+m', l-m),
a' <- region (l-m, l),
w <- back i (l, j-m'-2),
b' <- region (j-m', j),
energy = wkn *(stackenergy (i,l) + stackenergy(k,j)
              -stackenergy (i+m, l-m) - stackenergy(k+m', j-m'))
          + pbb*(m + m'-2)
```

Left to be defined are the interior structures *front*, *middle*, and *back*. For reasons of space, we only show the definition of *front*.

```
front j = idd <<< front' |||
         frd j <<< front' ~~~+ base
front' = pul <<< emptystrand |||
         mc <<< comps
```

This case takes care of a potentially dangling base from the b -helix, and if the remaining region is not empty, an arbitrary list of substructures (*comps*) is recognized. *idd*, *frd*, *pul*, and *mc* are the corresponding functions from the energy model.

[‡]To avoid an extra factor of n in time complexity, the energies of maximal length helices are also precomputed in table *stackenergy*. If the helices must be chosen shorter than maximal to avoid overlap, a correction term has to be subtracted. This explains the two negative terms in the energy computation.

[†]<http://bibiserv.techfak.uni-bielefeld.de/adp>

Sequence	n =	pknotsRE		pknotsRG	
		Time	MB	Time	MB
HIVRT	35	9 s	-	<1 s	2
t-RNA	75	18 min	6.6	17.3 s	8
TYMV	86	45 min	9.8	26.4 s	11
TMV	105	155 min	22.5	60.7 s	17
RNaseP	400	-	-	18.2 h	350

Table 2. Performance results for short sequences and comparison to *pknotsRE*; measured on an UltraSPARC CPU with 450MHz and 4GB main memory

The first clause (knot) chooses k, l in addition to i, j , computes m and m' using the precomputed maximal helix information, and passes these boundaries to the pseudoknot compartments. Methodically, this is a use of inherited attributes with the underlying tree grammar, and appears to be a novel technique in dynamic programming, at least in its grammar oriented tradition [Searls (1997); Lefebvre (1996); Rivas and Eddy (2000); Evers and Giegerich (2001)].

Practical evaluation

We tested our algorithm on all the examples supplied with the distribution *pknotsRE* and compared the resulting structures. Since *pknotsRG* computes a (canonized) subset of the structures considered by *pknotsRE*, we expected not to find any spurious pseudoknots in the prediction of 8 randomly chosen tRNAs. Indeed all tRNAs fold into a cloverleaf structure similar to those in the database. As an example of short pseudoknots we folded seven HIV-1-RT-ligands and obtained for all of them the identical structures as *pknotsRE* and the folding time was reduced from 9 to less than one second. Even for the 3' end of the *tymv* genome (86 nucleotides) we got an almost identical result, but taking only 26 seconds instead of the 45 minutes that *pknotsRE* required. *pknotsRG* also confirmed three small pseudoknots in prion protein mRNAs reported in Barette et al. (2001).

Clearly, we are able to fold sequences that are longer than *pknotsRE*'s limit of 130–140 nucleotides. E.g., a subsequence (250 nuc.) of the RNaseP RNA was folded in 53 minutes and the complete sequence (400 nuc.) in 18 hours. Furthermore the storage use did not exceed the theoretically expected bounds. These measurements are summarized in Figure 2.

Seq.	n =	pknotsRG		bestknot		glintron	
		Time	MB	Time	MB	Time	MB
E.c.	377	6:47 h	320	2:08 h	105		
T.th. r = 250 r = 220	419	13 h	417	3:18 h	142	9.6 min 7.2 min	110 109

Table 3. Performance results for long sequences and three variants of our algorithm; measured on an UltraSPARC CPU with 900MHz and 64GB main memory. Program *glintron* is described below; here it located a motif of maximum size r in the 419 nt T.th. sequence.

For a fair comparison, the reader should keep in mind that the extra time spent by *pknotsRE* is not strictly wasted: It is spent on assuring that the folding space of the input RNA sequence does contain pseudoknots with chained interacting helices of lower free energy than the reported structure. *pknotsRG* does not consider such structures and hence cannot make this assertion.

Some measurements for longer sequences, where no comparison with other programs is possible, are shown in Table 3, using three variants of our algorithm.

Bulges, triple crossing and kissing hairpins

Canonization Rule 1 can be relaxed to allow bulges inside the helices forming a pseudoknot. As long as their number (and hence the length difference of the two arms of a helix) is bounded by a constant, asymptotic efficiency is not affected.

Two examples of non-simple pseudoknots are shown in Figure 3. We can incorporate them into our algorithm adding the definitions

```
kiss = kss <<< a~~u~~b~~v~~a'~~w~~c~~x~~b'~~y~~c'
triple = trp <<< a~~u~~b~~v~~c~~w~~a'~~x~~b'~~y~~c'
```

Canonization can be applied as above, with Rule 3 becoming more sophisticated for the triple interaction case. This would yield an algorithm of runtime $O(n^6)$, bringing runtime back to the efficiency class of the Rivas/Eddy algorithm. But note that the space requirements remain $O(n^2)$. This is due to the fact that we now consider three interacting helices, but not arbitrary chains.

Folding long sequences

RNA folding *in vivo* as *in vitro* must be understood a hierarchical process, where small structures

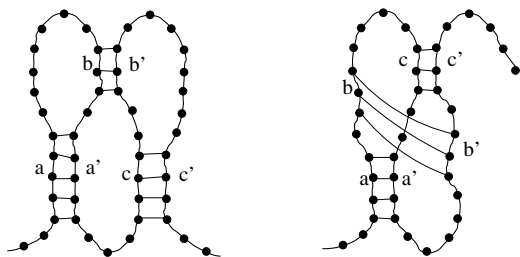


Fig. 3. Kissing hairpins (left) and triple helix interaction(right).

in close vicinity form first, and then combine to larger ones [Tinoco Jr and Bustamante (1999)]. The folding path becomes relevant, and the longer a sequence, the more unlikely it is that its folding path leads to a global energy minimum. In other words, the longer the sequence, the less reliable are the results of minimum free energy folding. *pknotsRG* gives us the possibility to test this using two fairly large structures including pseudoknots that have been proved experimentally. We considered two sequences: a RNaseP sequence from E.coli (419 NT) and a group I intron from *Tetrahymena thermophila* (377 NT). The mfe-structure found in each case was quite different from the “true” structure taken from the literature. We handcoded the experimental structures and evaluated their stability in our energy model. The result was striking: in both cases the experimental structure was significantly far from the possible minimum of free energy. So far in fact that it seems infeasible to detect these structures by scanning the space of near-optimal structures. This could be interpreted as the energy model being incorrect, but since it works well for short sequences, we suggest that this is an indication that the kinetics of folding already have a strong influence with this size of sequence, at least when pseudoknots are involved. Figure 4 shows these observations.

While we have achieved a considerable speedup for predicting small pseudoknotted structures, it seems that minimum free energy approach is not meaningful with the larger structures which it can handle algorithmically.

However, the situation changes when we are looking for particular structural motifs. Algorithm *pknotsRG* can serve as a generic template for developing matching algorithms based on thermodynamics for structures including pseudoknots.

Seq.	n =	experi- mental kcal/Mol	pknotsRG kcal/Mol	bestknot kcal/Mol	glintron kcal/Mol
E.c. m =	377 372	-104.33 372	-155.80 373	-117.03 375	-70.73 290
T. th. m=	419 419	-83.24 419	-142.50 419	-113.92 413	-49.71 217

Table 4. Energies computed for experimental structures and for structures predicted by three variants of our algorithm. m indicates the size of the motif from which the energy value is computed.

DERIVING THERMODYNAMIC MATCHERS

Thermodynamic matching

Algorithms for matching RNA structural motifs can be based on combinatorics, finding an arrangement of helices [Lisacek et al. (1994); Sagot and Viari (1997)]. In RNAMotif Macke et al. (2001), thermodynamic stability is one of several scoring functions that can be employed. Complex structures are built up in a tree like fashion comparable to the view adopted here. The motif search procedure is a combinatorial heuristic; unfortunately, no performance figures are given in Macke et al. (2001). Good scoring functions and many motif parameters are essential to control the search space, and to compensate the undesired effect of forcing a sequence into the motif structure that energetically favours a different structure. By contrast, a *thermodynamic matcher* delivers the energetically best (sub)structure that fits the predefined motif, by adjusting the folding algorithm to the specialized search space. Its usage is different: Rather than describing the motif as tightly as possible, one only specifies its general shape and lets thermodynamics control the folding. In our examples we only needed a single parameter. If the resulting structure is close to what we hope to find, this has the additional justification of being supported by the energy model.

Having an ADP description available for arbitrary structures including pseudoknots, there is a systematic way to derive thermodynamic matchers for specific motifs that contain pseudoknots. The method, in general, is based on formal language methods: Specializing the grammar that describes the class of all structures to one that describes the motif, while retaining all aspects that relate to en-

derived from *pknotsRG* finds the best group-1-intron core structure in a given sequence, based on the standard MFE model. We found that thermodynamic matching is much more flexible than combinatorial matching - there are fewer parameters and less danger of overfitting the motif to the data. Essentially, an overall bound on the motif size is all that is required for good results, and a motif can be developed by successive refinements. While comparative analysis remains to be the main technique to derive consensus structures, we expect that thermodynamic matchers as presented here can aid such work considerably.

Derivation of thermodynamic matchers for new motifs including pseudoknots is a systematic effort of moderate difficulty, but requires mastering the ADP technique. Researchers interested in such matchers (but not familiar with ADP) should consider the possibility to cooperate with bioinformatics students trained in this method within the framework of the virtual study project agency[†].

ACKNOWLEDGEMENT

We gratefully acknowledge the help of Dirk Evers, whose ADP code for folding unknotted structures served as a starting point for our implementation. We also thank Elena Rivas for discussing effects of canonization and Marc Rehmsmeier for carefully reading this manuscript.

This article was submitted to a major conference early in 2003. While the paper was positively commented for its novel ideas, it was not accepted ultimately. Meanwhile, the paper had leaked from the PC to outsiders. For this reason, the paper is reproduced here as submitted in early 2003 - including a number of typing errors for which we apologize.

REFERENCES

- Akutsu, T. (2000). Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics* 104, 45–62.
- Barette, I., G. Poisson, P. Gendron, and F. Major (2001). Pseudoknots in prion protein mRNAs confirmed by comparative sequence analysis and pattern searching. *Nucleic Acids Research* 29(3), 753–758.
- Cech, T. (1988). Conserved sequences and structures of group I introns: building an active site for RNA catalysis—a review. *Gene* 73, 259–271.
- Dennis, C. (2002). The brave new world of RNA. *Nature* 418, 122–124.
- Evers, D. and R. Giegerich (2001). Reducing the conformation space in RNA structure prediction. In *German Conference on Bioinformatics*, pp. 118–124.
- Giegerich, R. (2000a). Explaining and controlling ambiguity in dynamic programming. In *Proc. Combinatorial Pattern Matching*, pp. 46–59. Springer Verlag.
- Giegerich, R. (2000b). A systematic approach to dynamic programming in bioinformatics. *Bioinformatics* 16, 665–677.
- Giegerich, R. and C. Meyer (2002). Algebraic Dynamic Programming. In H. Kirchner and C. Ringeissen (Eds.), *Algebraic Methodology And Software Technology, 9th International Conference, AMAST 2002*, Saint-Gilles-les-Bains, Reunion Island, France, pp. 349–364. Springer LNCS 2422.
- Hofacker, I., W. Fontana, P. Stadler, L. Bonhoeffer, M. Tacker, and P. Schuster (1994). Fast folding and comparison of RNA secondary structures. *Monatshefte Chemie* 125, 167–188.
- Lefebvre, F. (1996). A grammar-based unification of several alignment and folding algorithms. In *Proceedings 4th ISMB*, pp. 143–154. AAAI Press, Menlo Park, CA, USA.
- Lisacek, F., Y. Diaz, and F. Michel (1994). Automatic identification of group I intron cores in genomic DNA sequences. *Journal of Molecular Biology* 235, 1206–1217.
- Lyngsø, R. B. and C. N. Pedersen (2000). Pseudoknots in RNA secondary structures. In *Proceedings of the fourth annual international conference on computational molecular biology*, pp. 201–209. ACM Press.
- Lyngsø, R. B. and C. N. Pedersen (2001). RNA pseudoknot prediction in energy based models. *Journal of Computational Biology* 7, 409–428.
- Lyngsø, R. B., M. Zuker, and C. N. Pedersen (1999). Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics* 15(6), 440–445.
- Macke, T., D. Ecker, R. Gutell, D. Gautheret, D. and Case, and R. Sampath (2001). RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Research* 29(22), 4724–4735.
- Rivas, E. and S. R. Eddy (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology* 285, 2053–2068.
- Rivas, E. and S. R. Eddy (2000). The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics* 16(4), 334–340.
- Sagot, M.-F. and A. Viari (1997). Flexible identification of structural objects in nucleic acid sequences: palindromes, mirror repeats, pseudoknots and triple helices. In A. Apostolico and J. Hein (Eds.), *Combinatorial Pattern Matching 97, Lecture Notes in Computer Science*, Volume 1264, pp. 224–246. Springer Verlag.
- Searls, D. (1997). Linguistic approaches to biological sequences. *CABIOS* 13(4), 333–344.
- Tabaska, J. E., R. B. Cary, H. N. Gabow, and G. D. Stormo (1998). An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics* 14(8), 691–699.
- Tinoco Jr, I. and C. Bustamante (1999). How RNA folds. *Journal of Molecular Biology* 293, 271–281.
- Turner, D., N. Sugimoto, and S. Freier (1988). RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry* 17, 167–192.

[†]<http://www.techfak.uni-bielefeld.de/bcd/Spa/>

-
- Wuchty, S., I. Fontana, W. Hofacker, and P. Schuster (1998). Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers* 49, 145–165.
- Zuker, M. and S. Sankoff (1984). RNA secondary structures and their prediction. *Bull. Math. Biol.* 46, 591–621.

Bisher erschienene Reports an der Technischen Fakultät
Stand: 2003-05-28

- 94-01** Modular Properties of Composable Term Rewriting Systems
(Enno Ohlebusch)
- 94-02** Analysis and Applications of the Direct Cascade Architecture
(Enno Littmann, Helge Ritter)
- 94-03** From Ukkonen to McCreight and Weiner: A Unifying View of Linear-Time Suffix
Tree Construction
(Robert Giegerich, Stefan Kurtz)
- 94-04** Die Verwendung unscharfer Maße zur Korrespondenzanalyse in Stereo
Farbbildern
(André Wolfram, Alois Knoll)
- 94-05** Searching Correspondences in Colour Stereo Images – Recent Results Using the
Fuzzy Integral
(André Wolfram, Alois Knoll)
- 94-06** A Basic Semantics for Computer Arithmetic
(Markus Freericks, A. Fauth, Alois Knoll)
- 94-07** Reverse Restructuring: Another Method of Solving Algebraic Equations
(Bernd Bütow, Stephan Thesing)
- 95-01** PaNaMa User Manual V1.3
(Bernd Bütow, Stephan Thesing)
- 95-02** Computer Based Training-Software: ein interaktiver Sequenzierkurs
(Frank Meier, Garrit Skrock, Robert Giegerich)
- 95-03** Fundamental Algorithms for a Declarative Pattern Matching System
(Stefan Kurtz)
- 95-04** On the Equivalence of E-Pattern Languages
(Enno Ohlebusch, Esko Ukkonen)
- 96-01** Static and Dynamic Filtering Methods for Approximate String Matching
(Robert Giegerich, Frank Hischke, Stefan Kurtz, Enno Ohlebusch)
- 96-02** Instructing Cooperating Assembly Robots through Situated Dialogues in Natural
Language
(Alois Knoll, Bernd Hildebrand, Jianwei Zhang)
- 96-03** Correctness in System Engineering
(Peter Ladkin)

- 96-04** An Algebraic Approach to General Boolean Constraint Problems
(Hans-Werner Gsgen, Peter Ladkin)
- 96-05** Future University Computing Resources
(Peter Ladkin)
- 96-06** Lazy Cache Implements Complete Cache
(Peter Ladkin)
- 96-07** Formal but Lively Buffers in TLA+
(Peter Ladkin)
- 96-08** The X-31 and A320 Warsaw Crashes: Whodunnit?
(Peter Ladkin)
- 96-09** Reasons and Causes
(Peter Ladkin)
- 96-10** Comments on Confusing Conversation at Cali
(Dafydd Gibbon, Peter Ladkin)
- 96-11** On Needing Models
(Peter Ladkin)
- 96-12** Formalism Helps in Describing Accidents
(Peter Ladkin)
- 96-13** Explaining Failure with Tense Logic
(Peter Ladkin)
- 96-14** Some Dubious Theses in the Tense Logic of Accidents
(Peter Ladkin)
- 96-15** A Note on a Note on a Lemma of Ladkin
(Peter Ladkin)
- 96-16** News and Comment on the AeroPeru B757 Accident
(Peter Ladkin)
- 97-01** Analysing the Cali Accident With a WB-Graph
(Peter Ladkin)
- 97-02** Divide-and-Conquer Multiple Sequence Alignment
(Jens Stoye)
- 97-03** A System for the Content-Based Retrieval of Textual and Non-Textual Documents Based on Natural Language Queries
(Alois Knoll, Ingo Glckner, Hermann Helbig, Sven Hartrumpf)

- 97-04** Rose: Generating Sequence Families
(Jens Stoye, Dirk Evers, Folker Meyer)
- 97-05** Fuzzy Quantifiers for Processing Natural Language Queries in Content-Based Multimedia Retrieval Systems
(Ingo Glöckner, Alois Knoll)
- 97-06** DFS – An Axiomatic Approach to Fuzzy Quantification
(Ingo Glöckner)
- 98-01** Kognitive Aspekte bei der Realisierung eines robusten Robotersystems für Konstruktionsaufgaben
(Alois Knoll, Bernd Hildebrandt)
- 98-02** A Declarative Approach to the Development of Dynamic Programming Algorithms, applied to RNA Folding
(Robert Giegerich)
- 98-03** Reducing the Space Requirement of Suffix Trees
(Stefan Kurtz)
- 99-01** Entscheidungskalküle
(Axel Saalbach, Christian Lange, Sascha Wendt, Mathias Katzer, Guillaume Dubois, Michael Höhl, Oliver Kuhn, Sven Wachsmuth, Gerhard Sagerer)
- 99-02** Transforming Conditional Rewrite Systems with Extra Variables into Unconditional Systems
(Enno Ohlebusch)
- 99-03** A Framework for Evaluating Approaches to Fuzzy Quantification
(Ingo Glöckner)
- 99-04** Towards Evaluation of Docking Hypotheses using elastic Matching
(Steffen Neumann, Stefan Posch, Gerhard Sagerer)
- 99-05** A Systematic Approach to Dynamic Programming in Bioinformatics. Part 1 and 2: Sequence Comparison and RNA Folding
(Robert Giegerich)
- 99-06** Autonomie für situierte Robotersysteme – Stand und Entwicklungslinien
(Alois Knoll)
- 2000-01** Advances in DFS Theory
(Ingo Glöckner)
- 2000-02** A Broad Class of DFS Models
(Ingo Glöckner)

- 2000-03** An Axiomatic Theory of Fuzzy Quantifiers in Natural Languages
(Ingo Glöckner)
- 2000-04** Affix Trees
(Jens Stoye)
- 2000-05** Computergestützte Auswertung von Spektren organischer Verbindungen
(Annika Büscher, Michaela Hohenner, Sascha Wendt, Markus Wiesecke, Frank Zöllner, Arne Wegener, Frank Bettenworth, Thorsten Twellmann, Jan Kleinlützum, Mathias Katzer, Sven Wachsmuth, Gerhard Sagerer)
- 2000-06** The Syntax and Semantics of a Language for Describing Complex Patterns in Biological Sequences
(Dirk Strothmann, Stefan Kurtz, Stefan Gräf, Gerhard Steger)
- 2000-07** Systematic Dynamic Programming in Bioinformatics (ISMB 2000 Tutorial Notes)
(Dirk J. Evers, Robert Giegerich)
- 2000-08** Difficulties when Aligning Structure Based RNAs with the Standard Edit Distance Method
(Christian Büschking)
- 2001-01** Standard Models of Fuzzy Quantification
(Ingo Glöckner)
- 2001-02** Causal System Analysis
(Peter B. Ladkin)
- 2001-03** A Rotamer Library for Protein-Protein Docking Using Energy Calculations and Statistics
(Kerstin Koch, Frank Zöllner, Gerhard Sagerer)
- 2001-04** Eine asynchrone Implementierung eines Microprozessors auf einem FPGA
(Marco Balke, Thomas Dettbarn, Robert Homann, Sebastian Jaenicke, Tim Köhler, Henning Mersch, Holger Weiss)
- 2001-05** Hierarchical Termination Revisited
(Enno Ohlebusch)
- 2002-01** Persistent Objects with O2DBI
(Jörn Clausen)
- 2002-02** Simulation von Phasenübergängen in Proteinmonoschichten
(Johanna Alichniewicz, Gabriele Holzschneider, Morris Michael, Ulf Schiller, Jan Stallkamp)
- 2002-03** Lecture Notes on Algebraic Dynamic Programming 2002
(Robert Giegerich)

- 2002-04** Side chain flexibility for 1:n protein-protein docking
(Kerstin Koch, Steffen Neumann, Frank Zöllner, Gerhard Sagerer)
- 2002-05** ElMaR: A Protein Docking System using Flexibility Information
(Frank Zöllner, Steffen Neumann, Kerstin Koch, Franz Kummert, Gerhard Sagerer)
- 2002-06** Calculating Residue Flexibility Information from Statistics and Energy based Prediction
(Frank Zöllner, Steffen Neumann, Kerstin Koch, Franz Kummert, Gerhard Sagerer)
- 2002-07** Fundamentals of Fuzzy Quantification: Plausible Models, Constructive Principles, and Efficient Implementation
(Ingo Glöckner)
- 2002-08** Branching of Fuzzy Quantifiers and Multiple Variable Binding: An Extension of DFS Theory
(Ingo Glöckner)
- 2003-01** On the Similarity of Sets of Permutations and its Applications to Genome Comparison
(Anne Bergeron, Jens Stoye)
- 2003-02** SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry
(Sebastian Böcker)