Prediction of RNA secondary structure including kissing hairpin motifs

Corinna Theis, Stefan Janssen, and Robert Giegerich

Faculty of Technology, Bielefeld University
33501 Bielefeld, Germany
robert@techfak.uni-bielefeld.de

Abstract. We present three heuristic strategies for folding RNA sequences into secondary structures including kissing hairpin motifs. The new idea is to construct a kissing hairpin motif from an overlay of two simple canonical pseudoknots. The difficulty is that the overlay does not satisfy Bellman's Principle of Optimality, and the kissing hairpin cannot simply be built from *optimal* pseudoknots. Our strategies have time/space complexities of $O(n^4)/O(n^2)$, $O(n^4)/O(n^3)$, and $O(n^5)/O(n^2)$. All strategies have been implemented in the program *pKiss* and were evaluated against known structures. Surprisingly, our simplest strategy performs best. As it has the same complexity as the previous algorithm for simple pseudoknots, the overlay idea opens a way to construct a variety of practically useful algorithms for pseudoknots of higher topological complexity within $O(n^4)$ time and $O(n^2)$ space.

1 Introduction

1.1 Biological relevance of pseudoknots in RNA structure

RNA is a chain molecule, the activated form of genetic information in all living organisms. Folding back onto itself, RNA forms secondary structure via base pairing of complementary nucleotides. Stacks of base pairs form helices, akin to the Watson-Crick helix of DNA, but with base pairs A-U, G-C, G-U, and occasionally some non-standard pairs. Ultimately, a tertiary (spatial) structure forms which is essential for biological function. *Pseudoknots* are structural motifs also defined via base pairing patterns, but, as they form late in the folding process, are generally considered as elements of tertiary structure.

Kissing hairpins are a common RNA folding motif belonging to the class of pseudoknots. The unpaired bases of a secondary structure build crossing base pairs by loop-loop interactions (the "kiss") and form a stable tertiary structure motif. Although these motifs have been known for over fifteen years, our understanding of kissing hairpins is still small. Especially viral genomes have been investigated for kissing hairpins, but also bacterial and eukaryotic ones. Researchers showed that kissing hairpins have important duties in a wide variety of RNA mediated processes. For example, they contribute extensively in stabilizing the structure and also play a role in viral plasmid DNA replication [5]

The original publication is available at www.springerlink.com
http://www.springer.com/computer/theoretical+computer+science/book/978-3-642-15293-1

or RNA synthesis [19]. Li et al. investigated in 2006 the mechanical unfolding of a minimal kissing complex [15]. They discovered that the loop-loop interaction is exceptionally stable.

1.2 RNA folding of nested structures

In RNA structure prediction, there is a dichotomy between prediction of *nested* and *pseudo-knotted* structures. The former is essentially a solved problem, whereas the latter is an active area of research. A structure holds a pseudoknot, if residues $i-j$ and $k-l$ form base pairs such that $i < k < j < l$. This situation is also called a *crossing* interaction. Without any crossing interaction, a structure is *nested*.

Nested structures can be naturally represented as trees, and they lend themselves to structure prediction in $O(n^3)$ time and $O(n^2)$ space. Early algorithms used a simple optimization criterion such as base pair maximization, while today's algorithms of practical relevance [27,14,17] use free energy minimization under an experimentally established thermodynamic model [18]. An improvement to $O(n^3/\log n)$ time for folding of nested structures has recently been contributed by Frid et al. [9], but this approach is not easily adapted to the established energy model. Recent progress in the field of nested structure prediction has been made mostly in the area of a more comprehensive analysis of the folding space [26,4], comparative prediction from multiple sequences [8], or trading the thermodynamic model for machine learning techniques [2].

1.3 Folding pseudoknots

Structures with pseudoknots are much more difficult to predict. Even under energy models much simpler than what we use in practice, prediction of the optimal pseudo-knotted structure has been shown to be NP-hard [16,1]. This has generated considerable interest in algorithms that solve the problem in polynomial time for restricted topologies of pseudoknots – see the review by Condon and Jabbari [7]. In an investigation of pseudoknot topologies [23], Rødland argues that the full topological complexity of pseudoknots is probably not needed in practical applications. For reasons of space, in the sequel we focus on those approaches which have resulted in realistic programs.

Pseudoknot folding using the established energy model was pioneered by Rivas and Eddy [22]. They presented an $O(n^6)$ time, $O(n^4)$ space algorithm for a fairly general class of pseudoknots. The high effort allows to fold only rather short sequences, and hence, the generality of the algorithm cannot really be exploited. A pragmatic approach was chosen by Reeder and Giegerich with the program *pknotsRG* [20]. They restricted the analysis to the class of *canonical simple recursive* pseudoknots, achieving $O(n^4)$ time, $O(n^2)$ space, and leading to a program widely used [1] today. The program *HotKnots* [21] uses a heuristics to assemble pseudoknots from low-energy helices.

[1] Counting over 200 downloads and over 4,000 submissions per year according to http://bibiserv.techfak.uni-bielefeld.de/statistics/

Quite recently, a new algorithm has been published in [6], but at the point of this writing, an implementation was not yet available. Our new approach presented here is an extension of the ideas used with *pknotsRG*, which we will review in necessary detail in Section 2.1.

1.4 Typology of structures

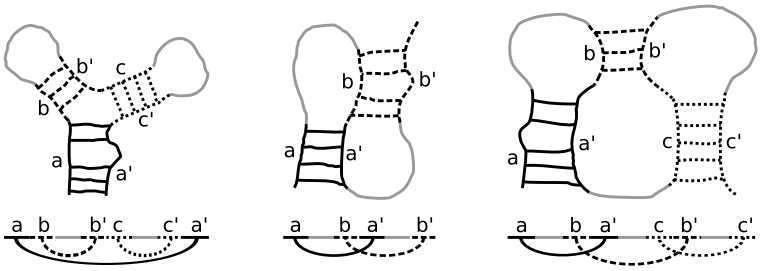

Fig. 1. Schematic representation of a nested structure (the Y shape), a simple pseudoknot, and a kissing hairpin motif. The bottom line shows the arrangement of helix parts mapped to the primary sequence, with arbitrary sequence in between.

Notation Dynamic programming over sequences leads to a decomposition of the given sequence into subwords, typically in all possible ways. Let $S = {}_0s_1 \ldots s_n$ be a sequence over the RNA alphabet $\{A, C, G, U\}$. The use of a fictitious 0-position at the start of S allows us to describe subwords by their bounding positions. For example, subword $(0, n)$ is S and subword $(2, 4)$ is ${}_2s_3s_4$. A subword (i, j) has length $j - i$ and splits seamlessly into subwords (i, k) and (k, j) for $i \leq k \leq j$. This convention avoids a lot of fiddling with ± 1.

We write $s = xyz$ to indicate that s is split into subwords x, y, z. The notation $s = {}_ix_ky_lz_j$ indicates, more concretely, that s is itself a subword of the overall input sequence S with boundaries i and j, and k, l denote the subword boundaries between x, y, z. If all boundaries are independent, a dynamic programming algorithm investigating all possible decompositions of this type has at least $O(n^4)$ steps, iterating over all $0 \leq i \leq k \leq l \leq j \leq n$.

Nested structures, simple pseudoknots, and kissing hairpins We use the notation axa' to indicate that subword a' is a reverse complement (under RNA rules) of a, and hence the two can form a helix. Using these conventions, Figure 1 sketches three types of RNA structures, together with their associated sequence decomposition. The first is a nested structure, the so-called Y-shape, the second a simple pseudoknot (sometimes called H-type), and the third is a kissing hairpin structure, which is our specific concern here. We shall reserve the word "pseudoknot" for simple pseudoknots here, to distinguish them from kissing hairpins. When we allude to pesudoknots with a more complex topology than these two classes, we shall explicitly say so.

To evaluate the folding energy of a kissing hairpin motif on subword s, we need to split $s = aubva'wcxb'yc'$. The subwords named u, v, w, x, y can attain arbitrary (sub)structures, so kissing hairpins (as well as pseudoknots) may be embedded within each other.

2 Three strategies for kissing hairpin prediction

2.1 The combined power of canonization rules and non-ambiguous dynamic programming

Canonization The algorithm of *pknotsRG* reduces computational complexity by imposing three canonization rules on the pseudoknots it considers:

Rule 1: In a helix $s = aua'$, a and a' are perfect helices.

Rule 2: In a helix $s = aua'$, a and a' extend towards each other maximally according to the rules of base pairing, except the following case:

Rule 3: With crossing helices as in $aubva'wb'$, Rule 2 might imply a negative length of v. We set $v = \varepsilon$ and both helices meet at an arbitrary position.

Note that these rules are imposed on pseudoknots only, the search space of nested structures remains untouched. The beneficial effect of these rules is that maximal helices of form ${}_iaza'_j$ can be precomputed, and a canonical split into a pseudoknot of form $s = aubva'wb'$ is uniquely characterized by four moving boundaries only, more precisely as $s = {}_iau_kbva'_lwb'_j$. This is the key to achieve $O(n^4)$ time, $O(n^2)$ space efficiency. For details, we refer to [20]. There, it is shown that while an optimal, pseudoknotted structure P may not satisfy the canonicity constraints, there is a near-optimal pseudoknot P_{can} which does. However, minimum free energy folding might deliver an unknotted structure U with free energy such that $E(P) \leq E(U) \leq E(P_{can})$. U will be returned without a hint to P_{can}, and hence to the potential existence of P. At this point, computing with canonical pseudoknots seems but another heuristic approach.

Semantic non-ambiguity A dynamic programming algorithm is called *semantically ambiguous* [10,11], if it examines an object of interest in its search space more than once. This typically leads to exponential explosion of redundant solution candidates. For finding a single, optimal solution in a dynamic programming algorithm, such redundancy does not matter, but it renders the algorithm useless for producing near-optimals. The *pknotsRG* program is implemented in a non-ambiguous way.

Combining canonicity with a non-ambiguous algorithm allows the program to return suboptimals. In particular, we can ask the best canonical pseudoknot from the near-optimal search space, even when the minimum free energy structure comes out unknotted. The best canonical pseudoknot P_{can} may be checked for potential extension to a non-canonical structure P of even lower energy. In this sense, the heuristic constraint of canonization appears tolerable. Our algorithms presented here adhere to the same idea. All considered structures are canonical, and there will be only one situation where a structure is considered twice.

2.2 Decomposition alternatives of the kissing hairpin motif

An elementary decomposition of a kissing hairpin leads to three helices ($a-a'$, $b-b'$, $c-c'$) with intervening sequences u, v, w, x, y, folded in arbitrary ways, with the overall arrangement $aubva'wcxb'yc'$. See Figure 2 for an illustration. Such a decomposition, in full generality, leads to 12 moving boundaries, and makes us resort to canonization. Rule 2 of our canonization constraints eliminates six moving boundaries – the inner endpoints of three helices, which are now fixed by the helix maximality rule. The remaining boundaries are the outer endpoints of the three helices. Iterating over these six boundaries would lead to an $O(n^6)$ time, $O(n^2)$ space strategy. Our goal is to do better than this.

Our key idea is the view of the kissing hairpin motif as an overlay of two simple pseudoknots (Figure 2). Given that we already know how to compute optimal simple pseudoknots for the overlapping subwords $aubva'zb'$ and $btcxb'yc'$, can we find their optimal overlay such that $z = wcx$ and $t = va'w$, thus defining the overall optimal decomposition into $aubva'wcxb'yc'$? Can we find its optimal energy as the sum from its two constituents?

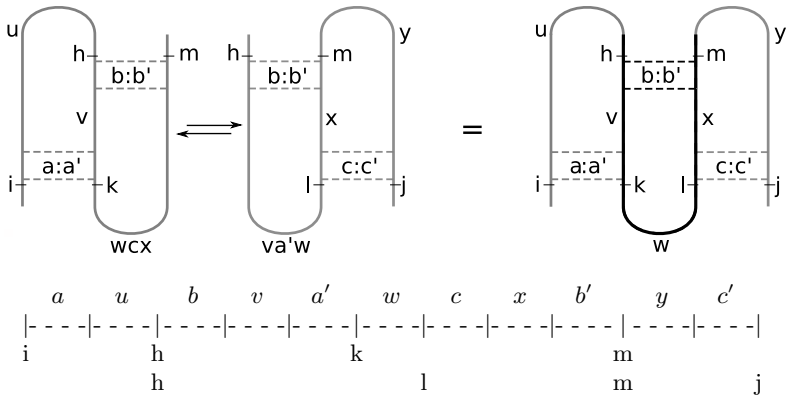

Fig. 2. The composition of two pseudoknots leading to a kissing hairpin motif with the overlay of parts of the sequence and the moving boundaries i, h, k, l, m, and j on top. The linear form of the sequence below shows 12 moving boundaries (vertical lines). With the canonization rules, only six boundaries (labeled lines) remain.

Simple as it seems, there is a problem. First, if $w = \varepsilon$, the optimal choice of a' (with respect to a and b') may conflict with the optimal choice of c (with respect to b and c'). Moreover, in the overlay, the energy contribution of the middle helix ($b-b'$) and the structure for v, w, and x embedded within both pseudoknots are accounted for twice, and must be subtracted from the energy sum of both parts. This violates the monotonicity requirement for dynamic programming known as Bellman's Principle: for the overlay, the energy function is non-monotonic, and

as a consequence, an optimal kissing hairpin motif may arise as an overlay of sub-optimal pseudoknots.

We will present three, increasingly complex strategies A, B, and C, such that their search spaces are properly included in the form $Searchspace_A \subseteq Searchspace_B \subseteq Searchspace_C \subset Searchspace_{KH}$. This relation will allow us to evaluate whether the expense for a more general strategy pays off in practice, but we will not be able to relate our results to an evaluation of the complete search space $Searchspace_{KH}$ of all (non-canonical) structures.

2.3 Strategy A – an $O(n^4)$ time, $O(n^2)$ space algorithm

Strategy A makes the optimistic assumption that at least one of the pseudoknots is the optimal structure for its underlying subword. This fixed, we choose the rest of the motif in the best possible way.

(1) For all subwords p, find the optimal pseudoknot such that $p = aubva'zb'$. Store results in a table of size $O(n^2)$.

(2) For all subwords s, split in all ways $s = pt$ and look up the optimal decomposition $p = aubva'zb'$.

(3) For all s of Step 2, use $s = auq$ and find the pseudoknot decomposition such that $q = brcxb'yc'$ and $r = va'w$, to complete the kissing hairpin decomposition $s = aubva'wcxb'yc'$. This pseudoknot must be chosen such that c lies strictly to the right of a', hence this is not, in general, the optimal pseudoknot over its underlying subword q. Record the decomposition of lowest free energy.

(4 - 6) Apply symmetric steps starting from an optimal choice for the right pseudoknot in the overlay.

(7) Choose lower energy value from (3) and (6); store it in a table of size $O(n^2)$.

The symmetry of (1-3) and (4-6) leads to the only case of ambiguity in our approach: If the two locally optimal pseudoknots make a perfect overlay as a kissing hairpin, this (optimal) structure will be found twice.

Efficiency: (1) takes $O(n^4)$ steps as with *pknotsRG*. (2) takes $O(n^3)$ steps, as the decomposition of p is already computed. (3) takes also $O(n^4)$, because it inherits $O(n^3)$ from Step 2 for all splits of s, which determine au and hence, the split auq. (Only) one extra factor of n arises from the split rc, which in turn determines the inner endpoints of helix $(c - c')$ due to the maximality rule, and hence implies the split yc'. (4-6) take $O(n^4)$ steps for symmetry reasons. (7) takes $O(n^2)$ steps. Postponing implementation details, we see that this yields an algorithm with $O(n^4)$ time, $O(n^2)$ space requirements.

Note that Strategy A does some redundant work – the right pseudoknot determined in Step 3 has already been considered as a (generally sub-optimal) pseudoknot in Step 1.

2.4 Strategy B – an $O(n^4)$ time, $O(n^3)$ space algorithm

Strategy B avoids the redundant work of Strategy A, and also enlarges the search space. We spend extra space in Step 1 to store results about sub-optimal pseudoknots.

(1) For $p = aubva'zb'$, and for each choice of b therein, we record the optimal choice of a'. Conversely, for each choice of a', we store the optimal choice of b. This requires two tables of size $O(n^3)$.
(2) For the kissing hairpin motif, we first choose a, b, b', and c', which costs $O(n^4)$, and use the stored information to optimally determine the other bounds for a' and c by lookup with $O(1)$.
(3) Unfortunately, the stored information may suggest that with an optimal choice, a' and c would overlap (and w have negative length).We correct this by a heuristic decision – selecting an a' further to the left and a c further to the right. This decision will also be based on precomputed information in order to retain a runtime of $O(n^4)$.
(4) We minimize over all cases considered.

The overall efficiency is $O(n^4)$ time and $O(n^3)$ space. Note that the search space here is more general than with strategy A, as neither pseudoknot needs to be optimal with respect to its underlying subword. This generalization lies with Step 1. In Strategy A, only the optimal choice of b within p is considered for overlay, while here, all possible choices of b are tried.

2.5 Strategy C – an $O(n^5)$ time, $O(n^2)$ space algorithm

Strategy C avoids the extra storage required by Strategy B. The necessary information is re-computed on demand, after choosing a, b, b' and c'. This increases runtime, but also allows us to avoid the heuristic decision when a' and c would overlap. For each choice of a', we compute the best choice of c strictly to its right. This threatens to raise time complexity to $O(n^6)$, but with a clever arrangement of computations and an extra table of size $O(n)$, we can keep it at $O(n^5)$.

The optimal choice of l with respect to (h, j) as a pseudoknot is a heuristics with respect to (i, j) as a kissing hairpin (see Figure 3). It assumes that $va'w$ can fold optimally. For the kiss, however, v and w can only fold individually, as they are separated by a', which is the partner of a. Thus, l need not be optimal for (i, j) as a kissing hairpin.

3 Algorithms

3.1 Algorithmic subtleties

Annotated energies When computing minimum free energies from pseudoknots, we will need to also record the internal boundaries of the given subword which achieved optimal energy. These will be data of the form (E, h, k). When we minimize over these tuples, we do this with a lexicographic ordering. This is consistent with mimimizing over energies alone. When two structures have the same energy, then the choice is arbitrary and remains unspecified.

Exact subword boundaries in the input decomposition Substructures have certain minimal sizes. For example, we forbid lonely pairs, i.e. helices of length 1. Therefore, in ${}_ia_kza'_j$, we do not iterate k over $i \leq k \leq j$, but only over $i+2 \leq k \leq j-2$. This does not affect the asymptotics, but saves substantial time in practice. The minimal subword sizes used are two base pairs for each helix, loop u and y have one unpaired base. Loop w has two single bases ($k+2 \leq l$). The size of loop v and x is ≥ 0, because we want to keep the possibility of coaxially stacking of the helices. With that, we get a minimal sequence length of 16 bases to form a kissing hairpin (see Figure 3).

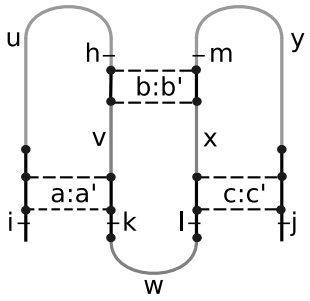

Fig. 3. The graphic shows the mandatory bases (black dots) of a kissing hairpin and the indices i, h, k, l, m, and j determining the start and end points of the helices (black tics). Gray regions u, v, w, x, and y can fold in an arbitrary way.

To be concrete in the following recurrences, we use the precise boundaries consistent with our implementation. But for understanding the essentials of the algorithms, the reader may choose to ignore them.

3.2 Pseudoknot-recurrence of *pknotsRG* – csrPK

Due to the canonization of *pknotsRG*, the calculation of a canonical simple recursive pseudoknot (csrPK) for a given subword needs two boundaries in addition to (i, j): h, the start position of the $b - b'$ helix, and k, the end position of the $a - a'$ helix. The recurrence of a csrPK for a subword (i, j) is:

$$\mathrm{csrPK}\,(i,j) = \min_{\substack{i+3 \,\leq\, h \,\leq\, j-8 \\ h+4 \,\leq\, k \,\leq\, j-4}} E_{\mathrm{csrPK}}\left({}_iau_hbva'_krb'_j\right)$$

The energy function E_{csrPK} makes use of a precomputed table to determine the inner endpoints of the helices in a unique, maximal and non-overlapping fashion. With these boundaries fixed, the energy value is the sum of stabilizing energies of both helices + energy contributions of the arbitrary folded regions u, v and

w + contributions from bases which dangle onto the helices from inside the csrPK + penalties for explicitly unpaired bases in front of u and b'. For later use, we adapt E_{csrPK} to additionaly store h and k, which can be retrieved by the functions $\text{boundary}_{\text{left}}$ and $\text{boundary}_{\text{right}}$.

3.3 Recurrences of Strategy A – csrKH$_A$

For Strategy A we make two strong assumptions. (1) Helices $a-a'$ and $b-b'$ of an optimal csrPK, starting at i and ending at m, can be adopted for the overall csrKH and thus determine the boundaries h and k. We can look up these values via the table csrPK. (2) The remaining boundary l, the starting point for the $c-c'$ helix, can be determined by using the energy of a second csrPK as an objective function. This second csrPK must start at h, end at j and have its end position of the left helix $b:b'$ at m, thus overlaying a part of the first csrPK:

$$\begin{aligned}
\text{left}\,(i,j) = &\underset{i+13\ \le\ m\ \le\ j-3}{min} E_{\text{csrKH}}\left({}_{i}au_{h}bva'_{k}w_{l}cxb'_{m}yc'_{j}\right), \quad \text{where}\\
&h = \text{boundary}_{\text{left}}\left(\text{csrPK}\,(i,m)\right),\\
&k = \text{boundary}_{\text{right}}\left(\text{csrPK}\,(i,m)\right),\\
&l = \text{boundary}_{\text{left}}\left(\underset{k+2\ \le\ d\ \le\ m-4}{min} E_{\text{csrPK}}\left({}_{h}bva'w_{d}cxb'_{m}yc'_{j}\right)\right)
\end{aligned}$$

A csrKH may alternatively arise from the opposite direction, i.e. an optimal csrPK on its right half overlaying a suboptimal csrPK at its left:

$$\begin{aligned}
\text{right}\,(i,j) = &\underset{i+3\ \le\ h\ \le\ j-13}{min} E_{\text{csrKH}}\left({}_{i}au_{h}bva'_{k}w_{l}cxb'_{m}yc'_{j}\right), \quad \text{where}\\
&l = \text{boundary}_{\text{left}}\left(\text{csrPK}\,(h,j)\right),\\
&m = \text{boundary}_{\text{right}}\left(\text{csrPK}\,(h,j)\right),\\
&k = \text{boundary}_{\text{right}}\left(\underset{h+4\ \le\ d\ \le\ l-2}{min} E_{\text{csrPK}}\left({}_{i}au_{h}bva'_{d}wcxb'_{m}\right)\right)
\end{aligned}$$

The optimal csrKH with Strategy A is:

$$\text{csrKH}_A\,(i,j) = min\left(\text{left}\,(i,j), \text{right}\,(i,j)\right)$$

3.4 Recurrences of Strategy B – csrKH$_B$

Since Strategy B has to store the optimal choice of a' for every given b for csrPKs on the left side and the optimal b for every given a' for csrPKs on the right side of the csrKH, we have to replace the function csrPK with lpk and rpk. A csrPK for a subword (i,j) can now be determined by minimizing over $lpk\,(i,h,j)$ and $rpk\,(i,k,j)$:

$$\begin{aligned}
lpk\,(i,h,j) = &\underset{h+4\ \le\ k\ \le\ j-4}{min} E_{\text{csrPK}}\left({}_{i}au_{h}bva'_{k}rb'_{j}\right)\\
rpk\,(i,k,j) = &\underset{i+3\ \le\ h\ \le\ k-4}{min} E_{\text{csrPK}}\left({}_{i}au_{h}bva'_{k}rb'_{j}\right)
\end{aligned}$$

An overlay of csrPKs from lpk and rkp might overlap in region w of the csrKH, when building it. We can overcome this obstacle in a heuristic way by introducing an artifical border ξ:

$$lpk_{\text{heuristic}}(i,h,j) = \min_{h+4 \le k \le \xi} E_{\text{csrPK}}\left({}_i a u_h b v a'_k r b'_j\right)$$
$$rpk_{\text{heuristic}}(i,k,j) = \min_{\xi \le h \le k-4} E_{\text{csrPK}}\left({}_i a u_h b v a'_k r b'_j\right)$$

Thus we can construct a csrKH with Strategy B by first iterating over the outer endpoints of helix $b - b'$, namely m and h. Second, we choose the energetically optimal combination of k and l by overlaying all csrPKs from $lpk(i,h,m)$ and $rpk(h,m,j)$, as well as their heuristic counterparts $lpk_{\text{heuristic}}(i,h,m)$ and $rpk_{\text{heuristic}}(h,m,j)$ to guarantee at least one feasible overlay:

$$\text{csrKH}_B(i,j) = \min_{\substack{i+13 \le m \le j-3 \\ i+3 \le h \le m-10}} E_{\text{csrKH}}\left({}_i a u_h b v a'_k w_l c x b'_m y c'_j\right), \quad \text{where}$$
$$k \in \text{boundary}_{\text{right}}\left\{lpk(i,h,m),\ lpk_{\text{heuristic}}(i,h,m)\right\}$$
$$l \in \text{boundary}_{\text{left}}\left\{rpk(h,m,j),\ rpk_{\text{heuristic}}(h,m,j)\right\}$$

3.5 Recurrences of Strategy C – csrKH$_C$

We start with Strategy C identical to Strategy B, by iterating over m and h. But instead of retrieving k and l from precomputed csrPK tables, we now also iterate k to determine a' and look up the optimal choice for l depending on k in a one dimensional table rpk:

$$\text{csrKH}_C(i,j) = \min_{\substack{i+13 \le m \le j-3 \\ i+3 \le h \le m-10 \\ h+4 \le k \le m-6 \\ l = \text{boundary}_{\text{left}}(rpk(k))}} E_{\text{csrKH}}\left({}_i a u_h b v a'_k w_l c x b'_m y c'_j\right)$$

When iterating over k, we go from right to left. Thus we have a growing subword (k,m). While shifting k one position to the left, the function $rpk(k)$ also determines the optimal csrPK that begins at h, ends at j, has its b' at m and its c somewhere in the subword (k,m). Since we temporarily store the results for $rpk(k)$, it can be calculated in $O(1)$ time. We just compare the existing result for the one letter shorter subword $rpk(k+1)$ with one new csrPK, whose boundaries are at $h, k+2, m, j$:

$$rpk(k) = min\left(E_{\text{csrPK}}\left({}_h b v a' w_{k+2} c x b'_m y c'_j\right), rpk(k+1)\right)$$

3.6 Implementation via algebraic dynamic programming

Alike *pknotsRG*, *pKiss* is implemented with the algebraic dynamic programming technique [12]. This makes it easy to add and combine different types of analysis. Currently, we compute optimal and suboptimal structures. We plan to add shape abstraction and computation of best knotted and un-knotted folding.

4 Evaluation of strategies A, B, and C

A piece of anecdotal evidence The RNA polymerase gene (gene 1) of the human coronavirus 229E is a good example for the usefulness of improved secondary structure prediction tools. Analyzing the genome of the human coronavirus, Herold and Siddell [13] guessed, that a "slippery site" together with an H-type pseudoknot acts as a frameshift inducing structure. Extensive mutational analyses showed that a kissing hairpin is required for high frequency frameshifts. Their work implied computer-assisted modeling, but prior prediction tools could not detect kissing hairpin motifs. *pKiss* finds the proper kissing hairpin.

Available test data Verified structures holding pseudoknots and kissing hairpins are rare. We collected a dataset of 61 pseudoknotted structures include 6 kissing hairpins, one "double" pseudoknot with topology $a\_b\_c\_d\_c'\_a'\_d'\_b'$ and 5 simple pseudoknots with nested sub-structures (see Appendix). The sequence length varies from 28 to 115 nt. The sequence types consist of viral ribosomal frame shifting or readthrough, mRNA, tmRNA, viral 3' UTR, ribozymes, signal recognition particle RNA [25], sequences with high affinity to HIV-1-RT [24] and viral RNA. These well-studied structures are subsequently called the true structures.

Comparison of the Strategies A, B, and C On 57 out of 61 sequences, Strategies A, B, and C agree. B finds a structure of lower energy than A in two cases, and C in the same two cases and two further ones. This is consistent with the hierarchy of search space inclusion, but the small disagreement is surprising.

Positive and negative test cases For a true positive prediction, we require the structure with the right topology in the right sequence position, but allow for a few missing base pairs (the price of canonization) or extra base pairs when they are consistent with the true structure. All 6 true kissing hairpins are precisely predicted by each strategy. Overall, 46 structures (75.4%) are correctly predicted while 15 sequences (24.6%) deviate from the true structure. These negative cases contain the complex pseudoknot which is beyond the class of kissing hairpins, but the helices actually predicted are correct. In seven cases, a kissing hairpin is predicted instead of a simple pseudoknot. One cannot exclude that this kissing hairpin is actually correct, but has not been detected before due to the lack of appropriate tools.

Further evaluations Comparing *pKiss* to the program by Rivas and Eddy brought little insight, as the program solves a more general problem and, as expected from their asymptotics, is much slower and greedy for space. Comparing *pKiss* to the most recent version of *HotKnots* [3] on our data set, we find the following: *HotKnots* currently provides four different parameter sets. Choosing the best prediction from those four in each case, it agrees with Strategy A in 3 out of our 6 positive test cases. On the larger data set of simple pseudoknots, there is more agreement between the methods. Execution time for a single parameter choice is generally lower than for *pKiss* by a factor of $3 - 6$. We have also

evaluated *pKiss* on random data and tested the robustness of predictions under varied energy parameters for kissing hairpin initiation. All evaluation data, as well as the first author's M.Sc. thesis, can be obtained from our website at http://bibiserv.techfak.uni-bielefeld.de/pkiss/.

5 Conclusion

Should the observations from our evaluation on sparse data generalize, interesting algorithmic perspectives open up. Strategy A evaluates a more complex motif than simple pseudoknots – without increasing asymptotic complexity. Unexpectedly, Strategy A performs best among A, B, and C – it is faster, agrees on the true positives, and has fewer false negatives. Closer inspection showed that it is always the left pseudoknot of the overlay which was chosen optimally. One may speculate that this is because the strategy is consistent with the hierarchic folding path during transcription. Boldly dropping the symmetric computation starting from the right pseudoknot reduces work in the innermost loop and may provide a speed-up factor close to 2.

The more exciting perspective is the extension of the overlay idea to more complex structures. A motif of four hairpins with two kissing interactions, for example, can be overlaid as $a\_b\_a'\_c\_b'\_c'$ and $b\_c\_b'\_d\_c'\_d'$. Using ideas of Strategy A, this can, again, be achieved in $O(n^4)$ time and $O(n^2)$ space! Additionally, alternative decompostions, say $a\_b\_a'\_c\_b'\_c'$ with $c\_d\_c'\_d'$ (a kissing hairpin overlaid with a simple pseudoknot) may be investigated, without raising the asymptotics. Furthermore, two such double kissing structures can form an overlay, and so on. It appears that one can construct a variety of practically useful, albeit increasingly heuristic, programs for pseudoknotted motifs of increasingly complex topologies within $O(n^4)$ time and $O(n^2)$ space.

Acknowledgement RG thanks A. Condon and H. Jabbari for discussion of the *pKiss* ideas in their early state.

References

1. T Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Appl Math*, 104(1–3):45–62, 2000.
2. MS Andronescu, AE Condon, HH Hoos, DH Mathews, and KP Murphy. Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, 23:19–28, 2007.
3. MS Andronescu, C Pop, and AE Condon. Improved free energy parameters for RNA pseudoknotted secondary structure prediction. *RNA*, 16(1):26–42, 2010.
4. CY Chan, CE Lawrence, and Y Ding. Structure clustering features on the Sfold Web server. *Bioinformatics*, 21(20):3926–3928, 2005.
5. KY Chang and I Tinoco. Characterization of a "kissing" hairpin complex derived from the human immunodeficiency virus genome. *Proc Natl Acad Sci USA*, 91(18):8705–8709, 1994.
6. HL Chen, AE Condon, and H Jabbari. An $O(n^5)$ Algorithm for MFE Prediction of Kissing Hairpins and 4-Chains in Nucleic Acids. *J Comput Biol*, 16(6):803–815, 2009.

7. AE Condon and H Jabbari. Computational prediction of nucleic acid secondary structure: Methods, applications, and challenges. *Theoretical Computer Science*, 410(4–5):294–301, 2009.
8. D Deblasio, J Bruand, and S Zhang. PMFastR: A New Approach to Multiple RNA Structure Alignment. *Lecture Notes in Computer Science*, 5724:49–61, 2009.
9. Y Frid and D Gusfield. A simple, practical and complete $O(n^3/\log n)$-time Algorithm for RNA folding using the Four-Russians Speedup. *Algorithms Mol Biol*, 5(1):13, 2010.
10. R Giegerich. Explaining and Controlling Ambiguity in Dynamic Programming. In *Proc. Combinatorial Pattern Matching*, volume 1848 of *Springer Lecture Notes in Computer Science*, pages 46–59. Springer, 2000.
11. R Giegerich and C Hoener zu Siederdissen. Semantics and Ambiguity of Stochastic RNA Family Models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(PrePrints), 2010.
12. R Giegerich, C Meyer, and P Steffen. A discipline of dynamic programming over sequence data. *Science of Computer Programming*, 51(3):215–263, June 2004.
13. J Herold and SG Siddell. An 'elaborated' pseudoknot is required for high frequency frameshifting during translation of HCV 229E polymerase mRNA. *Nucl Acids Res*, 21(25):5838–5842, 1993.
14. IL Hofacker, W Fontana, PF Stadler, SL Bonhoeffer, M Tacker, and P Schuster. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh Chem*, 125:167–188, 1994.
15. PTX Li, C Bustamante, and I Tinoco. Unusual mechanical stability of a minimal RNA kissing complex. *Proc Natl Acad Sci USA*, 103(43):15847–15852, 2006.
16. RB Lyngsø and CNS Pedersen. RNA Pseudoknot Prediction in Energy-Based Models. *J Comput Biol*, 7(3–4):409–427, 2000.
17. DH Mathews, MD Disney, JL Childs, SJ Schroeder, M Zuker, and DH Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc Natl Acad Sci USA*, 101(19):7287–7292, 2004.
18. DH Mathews and DH Turner. Prediction of RNA secondary structure by free energy minimization. *Curr Opin Struct Biol*, 16(3):270–278, 2006.
19. WJG Melchers, JGJ Hoenderop, HJ Bruins Slot, CWA Pleij, EV Pilipenko, VI Agol, and JMD Galama. Kissing of the two predominant hairpin loops in the coxsackie B virus 3' untranslated region is the essential structural feature of the origin of replication required for negative-strand RNA synthesis. *J Virol*, 71(1):686–696, 1997.
20. J Reeder and R Giegerich. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, 5(1):104, 2004.
21. J Ren, B Rastegari, AE Condon, and HH Hoos. HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. *RNA*, 11(10):1494–1504, 2005.
22. E Rivas and SR Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J Mol Biol*, 285(5):2053–2068, 1999.
23. EA Rødland. Pseudoknots in RNA Secondary Structures: Representation, Enumeration, and Prevalence. *J Comput Biol*, 13(6):1197–1213, 2006.
24. C Tuerk, MacDougal S, and L Gold. RNA pseudoknots that inhibit HIV type 1 reverse transcriptase. *Proc Natl Acad Sci USA*, 89(15):6988–6992, 1992.
25. FHD van Batenburg, AP Gultyaev, and CWA Pleij. PseudoBase: structural information on RNA pseudoknots. *Nucl Acids Res*, 29(1):194–195, 2001.

26. S Wuchty, W Fontana, IL Hofacker, and P Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, February 1999.
27. M Zuker and P Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucl Acids Res*, 9(1):133–148, January 1981.