BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 13 2013, pages 1607–1613
doi:10.1093/bioinformatics/btt188

Sequence analysis

Advance Access publication April 24, 2013

Structural RNA alignment by multi-objective optimization

Thomas Schnattinger[1,2], Uwe Schöning[1] and Hans A. Kestler[2,*]

[1]Institute of Theoretical Computer Science, Ulm University, 89069 Ulm, Germany and [2]Research Group Bioinformatics and Systems Biology, Institute of Neural Information Processing, Ulm University, 89069 Ulm, Germany

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: The calculation of reliable alignments for structured RNA is still considered as an open problem. One approach is the incorporation of secondary structure information into the optimization criteria by using a weighted sum of sequence and structure components as an objective function. As it is not clear how to choose the weighting parameters, we use multi-objective optimization to calculate a set of Pareto-optimal RNA sequence-structure alignments. The solutions in this set then represent all possible trade-offs between the different objectives, independent of any previous weighting.

Results: We present a practical multi-objective dynamic programming algorithm, which is a new method for the calculation of the set of Pareto-optimal solutions to the pairwise RNA sequence-structure alignment problem. In selected examples, we show the usefulness of this approach, and its advantages over state-of-the-art single-objective algorithms.

Availability and implementation: The source code of our software (ISO C++11) is freely available at http://sysbio.uni-ulm.de/?Software and is licensed under the GNU GPLv3.

Contact: hans.kestler@uni-ulm.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on October 24, 2012; revised on April 15, 2013; accepted on April 17, 2013

1 INTRODUCTION

The central role of RNA is that of being a messenger between transcription and protein translation. However, it is known that non-coding RNA molecules have a wide range of functions in the cell, like catalysis of chemical processes or in gene regulation, and it is assumed that RNA played a central role in the early stages of life (Doudna and Cech, 2002; Latchman, 2005). As the functionality of RNA molecules is not only defined by their primary sequences but also the interaction of their base pairs, their secondary structure is well conserved during evolution (Zuker and Sankoff, 1984). Based on this observation, RNA can be organized into families with similar sequences and secondary structures. The *Rfam database* (Gardner *et al.*, 2011) is a collection of RNA sequences, arranged in currently 1973 different families, each represented by a multiple alignment of its members. When the sequence similarity is too low, the construction of reliable sequence alignments is impossible (Gardner *et al.*, 2005). In this case, structural information can be used as another criterium to improve these alignments. Although much work has already been put into the development of sequence alignment methods, especially those for multiple sequences (Larkin *et al.*, 2007), RNA sequence-structure alignment can still be considered as an open problem (Gardner *et al.*, 2005).

RNA secondary structure consists of base pairs (i_k, j_k), $i_k < j_k$ (positions from 5′ to 3′) between nucleotides of an RNA sequence. Through stacking effects, these base pairs tend to form helices and various kinds of loops (see Fig. 1 for an example). For complexity reasons, we will restrict ourselves to non-crossing structures that forbid so-called pseudo-knots. This means that if we have two base pairs (i_1, j_1) and (i_2, j_2) with $i_1 < i_2$, it follows that either $i_1 < j_1 < i_2 < j_2$ or $i_1 < i_2 < j_2 < j_1$. There are various methods that predict a secondary structure from a single RNA sequence by free-energy minimization, like the dynamic programming algorithm of Zuker and Stiegler (1981). However, it is known that the prediction of the structure of related RNA sequences can be improved by comparative methods (Westhof *et al.*, 1996). Most structure prediction tools only calculate non-crossing structures, but there are algorithms that predict structures with pseudo-knots. Therefore, they follow a hierarchical approach of predicting the secondary structure first and the pseudo-knots subsequently (Tinoco and Bustamante, 1999), even if recent works show that tertiary interactions emerge early in the RNA folding process and suppress incorrect structures (Behrouzi *et al.*, 2012).

One RNA molecule can have the ability to fold into different 'suboptimal' structures, which differ from each other in their stability. McCaskill (1990) gives an efficient algorithm that calculates all secondary structures, together with their occurrence probability in a thermodynamic equilibrium. The result is a matrix of probabilities P, where an entry at position (i,j) reflects

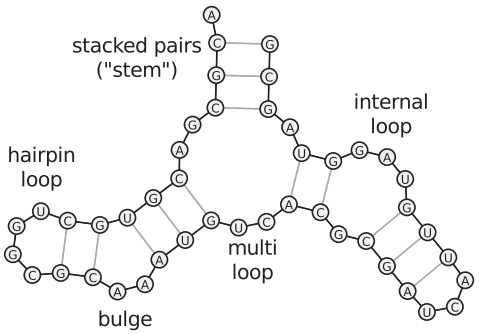

Fig. 1. Example of a RNA molecule. The primary sequence is the sequence of nucleotides (encircled), whereas the base pairings (in light grey) between nucleotides make up the secondary structure

*To whom correspondence should be addressed.

the probability of the bases at positions i and j to form a base pair. More precisely, P_{ij} is the sum of probabilities of all secondary structures that have a base pair between the nucleotides at positions i and j.

To our knowledge, Sankoff's algorithm (Sankoff, 1985) was the first to solve the alignment and folding of RNA simultaneously. In its pairwise version, it has a time and space complexity of $O(n^6)$ and $O(n^4)$, respectively, and as a consequence, it is often considered too expensive for a practical application. A simplified and more efficient variant of the Sankoff algorithm is Foldalign (Gorodkin *et al*., 1997; Havgaard *et al*., 2007), which uses a restricted energy model to reduce the time and space complexity, where Dynalign (Mathews, 2005; Mathews and Turner, 2002) features the full energy model, but it reduces the number of reasonable alignments by limiting the range of positions of matching nucleotides. Recently, Backofen *et al*. (2011) even presented an exact and efficient algorithm that uses sparsification to significantly enhance the runtime behaviour. Another popular implementation is *LocARNA* (Will *et al*., 2007), which is based on the work of Hofacker *et al*. (2004), and it incorporates structural information into the alignment process in the form of pre-calculated base pair probabilities. It finds an optimal sequence-structure alignment that maximizes an objective function, which is a weighted sum of a sequence alignment score and the sum of base pair probabilities of a consensus secondary structure. All these alignment methods depend on fixed weighting parameters for their sequence and their structure objectives, which have to be estimated or optimized. Depending on the type of RNA or the degree of structure conservation, a fixed weighting can lead to undesirable sequence-structure alignments. A change of these parameters can have a great influence on which consensus structure is considered as optimal.

We treat the sequence alignment and the consensus structure calculation as separate objectives, and we solve both problems simultaneously with a new multi-objective dynamic programming algorithm (Deb, 2001; Laux, 2005; Pareto, 1971). The result is not one single solution that is optimal in some sense, but a set of Pareto-optimal solutions. This technique has been applied successfully in different fields like the multi-objective routing problem (Sniedovich, 1988) or the knapsack problem with multiple criteria (Henig, 1983; Klamroth and Wiecek, 2000). In the bioinformatics area, Roytberg *et al*. (1999) construct a set of Pareto-optimal alignments of biological sequences by treating the number of gaps and the scores for (mis)matches as separate objectives. Taneda (2010) gives an evolutionary algorithm for pairwise RNA sequence alignment that incorporates RNA structure information to approximate a set of Pareto-optimal solutions. Although it also uses the theory of dominating vectors, it only calculates a rough approximation of the set of Pareto-optimal alignments. Apart from that, it depends on many parameters that influence the quality of approximation, as well as the runtime behaviour. We now present an algorithm that calculates the exact set of Pareto-optimal solutions.

2 METHODS

When the weighting between two conflicting objectives is not known, the weighting parameters have to be estimated or optimized in some way. After that the weighting parameters represent one fixed trade-off between the objectives, which also biases all subsequent solutions.

2.1 Multi-objective optimization

If more than one objective is to be optimized, the scoring can be treated as a vector-valued function, where the dimension is the number of objectives. Two different objectives can be conflicting, and in general, there is no solution that maximizes all objective functions. As there is no 'optimal solution' in a vector-valued context, the definition of optimality needs to be generalized.

Definition 1 (Dominating vector). *Let* $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ *and* $y = (y_1, \ldots, y_d) \in \mathbb{R}^d$. *We say* x *dominates* y *if* $x_i \geq y_i$ *for all* $1 \leq i \leq d$ *and* $x_j > y_j$ *for at least one* $1 \leq j \leq d$.

Definition 2 (Pareto-optimality). *Let* $M \subset \mathbb{R}^d$, $x \in M$. x *is Pareto-optimal in* M *if there is no other element* $y \in M$ *that dominates* x.

Now the problem of multi-objective optimization is the calculation of the largest subset of all possible solutions where every element is Pareto-optimal. This subset can contain multiple elements and is also known as the non-dominated set or the first Pareto-front (Laux, 2005). The task of finding the Pareto-optimal elements of a given finite set is called the maximal vector problem and has several solutions that are efficient in practice. As a special case, the computation of the Pareto-front of a set of 2D vectors takes linear time (Godfrey *et al*., 2007).

First, we show that every solution, which can be found by a mono-objective optimization, which maximizes a weighted sum of the objectives, is also Pareto-optimal.

Theorem 1. *Let* $M \subset \mathbb{R}^d$ *be some solution set, and let* f_i *be a family of objective functions. A solution* $m \in M$ *that maximizes*

$$\sum_i \omega_i f_i(m) \underset{m \in M}{\longrightarrow} \max \tag{1}$$

for any given weights $\omega_i > 0$ is also Pareto-optimal.

Proof. Let $m \in M$ be optimal in respect of (1). Assume m is not Pareto-optimal; therefore, there must exist an $y \in M$ that dominates m, i.e. $\forall i : f_i(y) \geq f_i(m)$ and $\exists j : f_j(y) > f_j(m)$. Without loss of generality be $f_1(y) > f_1(m)$. It follows that $\sum_i \omega_i f_i(y) > \sum_i \omega_i f_i(m)$, which contradicts the fact that m maximizes (1).

This result implies that our method is a generalization of any mono-objective method (compare Ehrgott, 2000). The set of supported solutions (i.e. there exist weighting parameters ω_i for which the solution is optimal) lies on the convex hull of the Pareto set. Restriction on these solutions may lead to non-favourable decisions (Klamroth and Wiecek, 2000). On the other hand, the elements from the Pareto-front represent every possible trade-off between the different objectives: one cannot get better in one without getting worse in another objective.

2.2 A multi-objective dynamic programming algorithm

The secondary structure of functional RNAs is often more conserved than their sequence; therefore, it makes sense to use the secondary structure as another criterion to obtain good alignments. A sequence-structure alignment of two RNA sequences A and B consists of a sequence alignment R and a compatible consensus secondary structure S, which means that both sequences can fold into the same structure S. More precisely, $(j, l) \in R$ means that the nucleotides A_j and B_l are matched together. $(i, j; k, l) \in S$ means $(i, k) \in R$ and $(j, l) \in R$ are matched nucleotides, and additionally, A_i is base paired with A_j, and B_k is base paired with B_l. We formulate two separate objectives that assign a score to a sequence-structure alignment. The first one in Equation (2) is the well-known objective function for sequence alignment algorithms: the sum

of penalties for the gaps in the alignment and of scores for (mis-)matched unpaired nucleotides.

$$f_{seq}(R,S) = \gamma \cdot N_{gap} + \sum_{\substack{(i,k)\in R \\ \text{unpaired}}} \sigma(A_i, B_k) \tag{2}$$

The gap penalty γ should be negative, where the scoring function σ for aligned nucleotides should satisfy $\sigma(a,a) \geq 0$ and $\sigma(a,b) \leq 0$ for $a \neq b$. Note that Equation (2) is a weighted sum itself. To be completely independent of weighting parameters, we could introduce a third objective function, but this would again result in a higher runtime complexity. The second objective function in Equation (3) is the sum of base pair scores for all aligned base pairs in the consensus secondary structure S.

$$f_{str}(R,S) = \sum_{(ij;kl)\in S} \Psi^A_{ij} + \Psi^B_{kl} \tag{3}$$

The score Ψ^A_{ij} for the base pair (A_i, A_j) is defined analogously to Hofacker *et al.* (2004), and it is proportional to the logarithm of the respective entry in the base pair probability matrix P^A_{ij} (McCaskill, 1990).

$$\Psi^A_{ij} = \begin{cases} \log(P^A_{ij}/p_{min}), & \text{if } P^A_{ij} \geq p_{min} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

In the first version of our algorithm (Schnattinger *et al.*, 2012), we used the recursive definition of Hofacker *et al.* (2004) for the prototypical implementation. In a more sophisticated algorithm by Will *et al.* (2007), the runtime and memory requirements were greatly improved. Therefore, we adapt this new formulation to fit the multi-objective problem definition. Let now $S_{i,j;k,l}$ be the set of Pareto-optimal scoring vectors of alignments of the two subsequences $A[i+1..j]$ and $B[k+1..l]$.

$$S_{i,j;k,l} = \textit{Pareto-max of}: \begin{pmatrix} \{s+\vec{\gamma} : s \in S_{i,j-1;k,l}\} \cup \\ \{s+\vec{\gamma} : s \in S_{i,j;k,l-1}\} \cup \\ \{s+\vec{\sigma}(A_j,B_l) : s \in S_{i,j-1;k,l-1}\} \cup \\ \bigcup_{h,q} \left\{ \begin{array}{c} s+d: \\ s \in S_{i,h-1;k,q-1}, \\ d \in D_{h,j;q,l} \end{array} \right\} \end{pmatrix} \tag{5}$$

$$D_{i,j;k,l} = \left\{ s + \vec{\Psi}^A_{i,j} + \vec{\Psi}^B_{k,l} : s \in S_{i,j-1;k,l-1} \right\} \tag{6}$$

with the initializations

$$S_{i,i;k,l} = \{\vec{\gamma}(l-k)\}, \quad \text{for} \quad l > k \tag{7}$$

$$S_{i,j;k,k} = \{\vec{\gamma}(j-i)\}, \quad \text{for} \quad j > i \tag{8}$$

$$S_{i,i;k,k} = \{\vec{0}\}. \tag{9}$$

Equation (5) takes the Pareto-optimal solutions from all candidate solutions, which originate from the union of four major sets. Each one results from reducing the problem of finding Pareto-optimal sequence-structure alignments of two sequences to smaller subproblems. The first one takes all elements of $S_{i,j-1;k,l}$ and adds a gap penalty $\vec{\gamma}$ to them. This stands for the insertion of a gap into the sequence B directly after position l, which is matched with A_j. Analogously, the second set describes the scoring vectors of all alignments that have B_l matched to a gap in A after position j. As $S_{i,j-1;k,l-1}$ is the set of alignments of the subsequences $A[i+1..j-1]$ and $B[k+1..l-1]$, the third set of Equation (5) contains all Pareto-optimal solutions to the alignment of $A[i+1..j]$ and $B[k+1..l]$, which have A_j matched (or mismatched) to B_l. The scoring of aligning these two nucleotides is done by the function $\vec{\sigma}$. The fourth part is the union of all candidate solutions, which have a base pair (h,j) in A and (q,l) in B. As a result, a solution vector is the sum of the solution vectors of two partial problems: the alignment of the subsequences $A[i+1..h-1]$ and $B[k+1..q-1]$, and of the subsequences $A[h..j]$ and $B[q..l]$ with the base pair $(h,j;q,l)$ as an additional condition. Therefore, Equation (6) defines $D_{i,j;k,l}$ to be the set of Pareto-optimal solution vectors of the alignments of the subsequences $A[i..j]$ and $B[k..l]$ with the condition that A_i is matched to B_k, A_j is matched to B_l, and these enclosing nucleotides are base paired. Obviously, it is required that both RNAs can have base pairs at these positions; hence, $P^A_{ij} > 0$ and $P^B_{kl} > 0$. Note that we do not need the Pareto-max operator here, as $S_{i,j-1;k,l-1}$ does not contain dominated solutions. Finally, Equations (7–9) specify the trivial base cases of the recursion.

We want to compute both scoring functions for every solution; hence, each element in the matrices S and D is a vector with two components: the sequence score and the structure score of this solution. Therefore, we implement a 2D scoring function with the first component containing the sequence score f_{seq} and the second component containing the structure score f_{str}. Thus, $\vec{\gamma}$ and $\vec{\gamma}(l)$ are the vector-valued gap penalties for the insertion of a gap of length 1 and l, which in our case is defined by $\vec{\gamma}(l) := (\gamma \cdot l, 0)$. The insertion of a gap into an existing intermediate solution should increase its sequence score by γ and leave the structure score untouched, which means that $\vec{\gamma}$ can simply be added to the scoring vector of the intermediate solution. Analogously, the scoring function for two aligned nucleotides a and b is $\vec{\sigma}(a,b) := (\sigma(a,b), 0)$. On the other hand, the vector-valued structure score is $\vec{\Psi}^A_{ij} := (0, \Psi^A_{ij})$, leaving the sequence score unchanged when added to the scoring vector of an intermediate solution.

The number of Pareto-optimal solutions to the sequence-structure alignment problem can be large. To maintain practicability, it is crucial to optimize the runtime and memory requirements of this algorithm. For example, S does not have to be calculated for every combination of i and k, as only those parts of the matrix are needed with either $(i,k) = (0,0)$ or A_i and B_k are positions where a base pair can start. The set of Pareto-optimal scoring vectors of global alignments of A and B is $S_{0,|A|;0,|B|}$. The dynamic programming matrix contains the Pareto-optimal scoring vectors for the solutions but not the sequence-structure alignments themselves. To get the desired alignments, a backtracing procedure is used in a second phase of the algorithm, which works as follows. For every scoring vector a, Equations (5) and (6) are used to figure out how this solution was calculated by simulating all steps that may have led to a. This is repeated recursively until one of the base cases from Equations (7–9) is encountered.

2.3 Proof

The recursive formulation of the dynamic programming algorithm is similar to the typical mono-objective formulation, except for the elements being vectors instead of numbers. To show the correctness of a dynamic programming algorithm, the problem must satisfy Bellman's principle of optimality (Bellman, 1957). Henig (1985) showed that the principle of optimality holds for multi-criteria decision problems if the monotonicity criterion is fulfilled. This means that if $a, b \in S_{i,j;k,l}$ are solutions to the same subproblem and a dominates b, and the same 'decision' $f(a)$ and $f(b)$ is applied to them, their dominance property must be preserved; hence, $f(a)$ must also dominate $f(b)$. We will show the correctness of our algorithm by showing that (i) for each subproblem (i,j,k,l), all Pareto-optimal solutions are generated and (ii) Bellman's principle of optimality holds; hence, no Pareto-optimal solution is lost by applying the Pareto-max operator during the algorithm.

THEOREM 2. *Equations (5) and (6), together with the initializations of Equations (7–9) compute the set of Pareto-optimal sequence-structure alignments.*

PROOF. Suppose that all smaller subproblems have already been calculated. Without the Pareto-max operator, Equations (5–9) produce the set of all possible sequence-structure alignments of A and B. The Pareto-set thereof is the desired result. We will now show that the removal of dominated solutions from the sets during the algorithm does not prevent

any Pareto-optimal solution to be computed in a future step. Therefore, we first show that a dominated intermediate solution can not be part of a Pareto-optimal solution. This argument covers the first three cases of Equation (5). Without loss of generality, we consider the first case, cases two and three follow analogously.

First case. Let $x, y \in S_{i,j-1;k,l}$ be two scoring vectors of the form $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$, with x dominating y, that is $x_i \geq y_i$ for $1 \leq i \leq d$ and $x_j > y_j$ for at least one $1 \leq j \leq d$. Furthermore, let $\vec{\gamma} = (\gamma_1, \ldots, \gamma_d)$ be the d-dimensional gap penalty score. Let now be $\hat{x} = x + \vec{\gamma} = (x_1 + \gamma_1, \ldots, x_d + \gamma_d)$ and $\hat{y} = y + \vec{\gamma} = (y_1 + \gamma_1, \ldots, y_d + \gamma_d)$. It is clear that $\hat{x}_i = x_i + \gamma_i \geq y_i + \gamma_i = \hat{y}_i$ *for* $1 \leq i \leq d$ and $\hat{x}_j = x_j + \gamma_j > y_j + \gamma_j = \hat{y}_j$ for at least one $1 \leq j \leq d$, which immediately implies that $\hat{x}$ dominates $\hat{y}$; hence, the monotonicity criterion holds. As a consequence, no Pareto-optimal solution is lost in the computation of $S_{i,j;k,l}$, if dominated scoring vectors are eliminated in their partial solutions in the first three cases.

Last case. For any fixed h and q be $v, w \in S_{i,h-1;k,q-1}$, $v = (v_1, \ldots, v_d)$ and $w = (w_1, \ldots, w_d)$ with v dominating w, and $x, y \in D_{h,j;q,l}$, $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ with x dominating y. Then the scoring vectors $(v + x)$, $(v + y)$, $(w + x)$ and $(w + y)$ are among the generated solutions for $S_{i,j;k,l}$. From the fact that v dominates w, it follows that $(v + x)$ dominates $(w + x)$, and $(v + y)$ dominates $(w + y)$, as shown earlier in the text. Also x dominates y; thus, $(v + x)$ dominates $(v + y)$ and $(w + x)$ dominates $(w + y)$. After the application of the Pareto-max operator, from these four produced solutions, only $(v + x) \in S_{i,j;k,l}$ survives in the set, and v and x were the non-dominated vectors. Therefore like aforementioned, by removing dominated solutions from intermediate results, no Pareto-optimal solutions are lost.

2.4 Implementation

The runtime and space requirements depend not only on the length of the input RNA sequences A and B, as it would be the case in the mono-objective algorithm, but also on the number K of Pareto-optimal solutions. This number depends on the sequence and structure conservation of the RNAs. If, for example, A (or B) does not have any secondary structure (i.e. $P^A = 0$), then the fourth case of Equation (5) always evaluates to the empty set. This means that for every (i, j, k, l), there will be no solution with a positive structure score. As a result, the number of Pareto-optimal solutions will be $K = 1$. In the other extreme, A and B have many suboptimal secondary structures; therefore, P^A and P^B have many non-zero entries. This will result in many sequence-structure alignments with different structure scores, and most likely in many non-dominated, hence, Pareto-optimal solutions.

We want to determine this number K (as a function of the sequence length) experimentally for the given 2D objective function. Therefore, we randomly chose 11 000 pairs of RNAs up to length 350 from the Rfam database (Gardner *et al.*, 2011), with each pair being from the same RNA family and differing no more in length than 10 nt. For each pair, we ran our algorithm and computed the size of the largest set of Pareto-optimal solutions among all intermediate solutions $S_{i,j;k,l}$. The experiment showed that the average quotient of the number of Pareto-optimal solutions and the mean sequence length of the input sequences is 1, and we conclude that K must be in $O(n)$. See Fig. 2 for a graphical result of this experiment.

The algorithm consists of two major parts. First, the set of Pareto-optimal scoring vectors $S_{0,|A|;0,|B|}$ is calculated using the recursion of Equations (5–9). The runtime complexity of a dynamic programming algorithm is the size of the table, which in our case is $O(n^4)$, times the runtime it takes to compute one entry. We saw earlier that the Pareto-max operator for 2D data has a runtime proportional to the number of candidate elements. Assuming that the largest set $S_{i,j;k,l}$ has K elements, it becomes clear that the first three cases of Equation (5) produce at most three K elements. Will *et al.* (2007) showed that, on average, the number of combinations (h,q) in the fourth case is constant. The elements in this fourth set are the sums of each combination of Pareto-optimal solutions of two smaller subproblems; hence, its size is in the order of K^2 in the worst case. This gives an overall runtime of $O(n^4 \cdot K^2)$. In practice, the runtime can be strongly improved by restricting the alignments in the following way. If we demand from each aligned pair (A_i, B_k) that the two nucleotides must be at most M positions apart, i.e. $|i - k| \leq M$, and furthermore that for matched base pairs $(i, j; k, l)$ the span of the two involved base pairs $|(j - i) - (l - k)| \leq M$ differs by no more than M, then the runtime can be reduced by another factor n^2 to $O(n^2 \cdot K^2)$ for fixed M (Hofacker *et al.*, 2004). Notice, however, that M has to be at least the length difference of A and B.

For fixed i and k, the calculation of $S_{i,j;k,l}$ depends only on entries of the form $S_{i,j';k,l'}$ and on entries from D. Therefore, after every iteration of i and k, the matrix S can be discarded, and only D has to be stored. This reduces the space complexity significantly from $O(n^4)$ to $O(n^2)$ in the mono-objective case (Will *et al.*, 2007). Therefore, in phase one of the multi-objective algorithm, we need $O(n^2 \cdot K)$ space to store S and D and another $O(K^2)$ for the calculation of the current matrix entry.

In the second phase of the algorithm, for every solution $s \in S_{0,|A|;0,|B|}$, a backtracing procedure is executed that recovers the actual alignment and the consensus secondary structure from its solution vector s and the dynamic programming matrices S and D. In the mono-objective algorithms, the backtracing walks once n steps through the matrix towards the optimal solution, and every step takes constant time. Even though some entries of the matrix have to be recalculated, the time and space consumption for this phase is negligible. In our case, we have to be more careful because larger parts of the matrix may have to be recalculated. As we expect a larger number of Pareto-optimal solutions, some maybe similar to each other, many entries will have to be recalculated multiple times. For this reason, we implemented the following hybrid approach. As described, we discard the intermediate solutions in the first part of the algorithm after every iteration of i and k. In the backtracing step, however, we memoize all recalculated entries and thus avoid redundant calculations.

This gives us a good time-space trade-off in practice, as we can see in Figure 3 showing the memory consumption of the three approaches, as well as their runtime by an example of two ykoK leader RNA

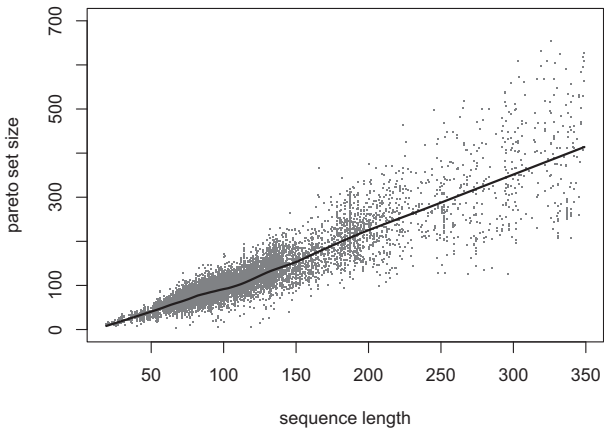

Fig. 2. The size of the set of Pareto-optimal solutions as a function of the mean sequence length. Each of the 11 000 grey dots is the size of the largest set of Pareto-optimal solutions of all subproblems $S_{i,j;k,l}$ of one experiment. The black line is an interpolation using the locally weighted regression and smoothing scatter plots (lowess) method of (Cleveland, 1979)

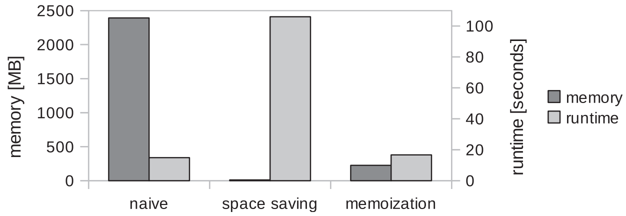

Fig. 3. Comparison of runtime and space requirement for the alignment of two ykoK leader RNA sequences of length 172 and 177, which results in a set of 157 Pareto-optimal solutions. Version 1 ('naïve') completes after 14.9 s, but it requires 2.3 GB memory. Version 2 ('space saving') needs only 12 MB memory, but it has a runtime of 106 s. With memoization, version 3 needs 260 MB and completes after 16.7 s. Experiments were conducted on an AMD Opteron(tm) 2431 processor with 2.4 GHz and 32 GB random access memory

sequences. The naïve approach that stores the whole DP matrix ('naïve') is the fastest, but it requires the most memory, as analysed earlier in the text. By discarding the intermediate solutions after each iteration of i and k ('space saving'), only a fraction of the space of the naïve approach is needed, but the recalculation of the missing entries in phase two leads to a poor runtime behaviour. In the hybrid approach ('memoization'), phase one only needs as little space as version 1. However, in phase two, the memoization of recalculated missing entries increases the memory consumption, but it leads to a much better runtime.

3 RESULTS

3.1 Alignment of two tRNAs

We align two RNA sequences using the well-established tool *LocARNA* (Will *et al.*, 2007) and compare the results to our multi-objective algorithm. Sequence *A* is *tRNA-Pro* from *Arabidopsis thaliana chromosome 2* (6810–6881) and sequence *B* is *tRNA-Lys* from *Marchantia polymorpha mitochondrion* (166035–166107), both taken from the *Rfam database* (Gardner *et al.*, 2011).

LocARNA calculates, according to its mono-objective function, the following sequence-structure alignment, correctly suggesting the acceptor-, D- and T-arm of the expected typical tRNA cloverleaf structure (Fig. 4, Solution 1):

```
GGGCAUUUGGUCUAG-UGGUAUGAUUCUCGCUUAGGGUGCGAGAGGUCCCGAGUUCAAUUCUCGGAAUGCCCC
GGGUGUAUAGCUCAGUUGGUAGAGCAAUAGGCUUUUAACUUAAAGGUCGCAGGUUCAAGUCCUGCUAUACCCA
(((((((..(((.........)))........................(((((.......)))))))))))).
```

If we recalculate the minimum free energy of these two RNA structures, which is a measure of stability, using *RNAeval* (Hofacker *et al.*, 1994), we get $-15.4\,\frac{kcal}{mol}$ for *A* and $-18.6\,\frac{kcal}{mol}$ for *B*. Our algorithm, in contrast, calculates a set of 37 Pareto-optimal sequence-structure alignments regarding the objective functions from Equations (2) and (3). Among them is the solution aforementioned, but another one is the following (Fig. 4, Solution 2):

```
GGGCAUUUGGUCUAG-UGGUAUGAU--UCUCGCUUAGGGUGCGAGAGGUCCCGAGUUCAAUUCUCGGAAUGCCCC
GGGUGUAUAGCUCAGUUGGUAGAGCAAUAGGCUUUUAACUUAAAGGUC--GCAGGUUCAAGUCCUGCUAUACCCA
(((((((..(((.........)))....(((((........)))))....(((((.......)))))))))))).
```

The additional stabilizing stem–loop structure of the anticodon arm leads to minimum free energies of $-24.0\,\frac{kcal}{mol}$ for *A* and $-20.1\,\frac{kcal}{mol}$ for *B*. Obviously, the mono-objective tool scores the gap penalty higher than the gain for the additional loop, and thus decides against the gaps, where our method leaves this final decision to the user. Figure 4 shows a plot of the Pareto-optimal scoring vectors. Here, each point represents one alignment including its structure information. The horizontal axis shows the sequence score, i.e. the sum of gap penalties and base match scores. The vertical axis represents the structure score, that is the sum $\Psi^A_{ij} + \Psi^B_{kl}$ of all matched base pairs (A_i, A_j) and (B_k, B_l). The two solutions that belong to the aforementioned sequence-structure alignments are highlighted. The different consensus secondary structures suggested by these two solutions can be observed in Figure 5.

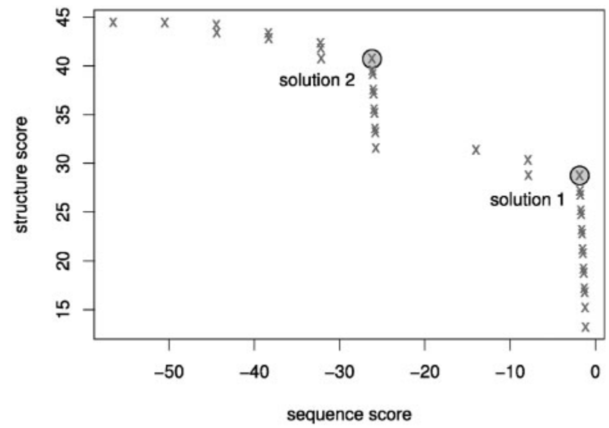

Fig. 4. 2D scatter plot of the Pareto-optimal alignments of two tRNA sequences, with the sequence score f_{seq} on the horizontal and the structure score f_{str} on the vertical axis

3.2 Benchmark

As a benchmark, we use the dataset 2 of the *BRAliBase II* benchmark (Gardner *et al.*, 2005), which consists of 117 pairwise reference alignments of tRNA sequences, as well as a dataset of 324 pairwise alignments of per cent identity balanced tRNA and 5S rRNA sequences published by Dowell and Eddy (2006) (Supplementary Material). We compare the results of the mono-objective structural alignment tools *LocARNA* (Will *et al.*, 2007) and *PMcomp* (Hofacker *et al.*, 2004), on which the first version of our algorithm was based, the multi-objective evolutionary algorithm *Cofolga2mo* (Taneda, 2010), and the state-of-the-art sequence alignment program *Clustal Omega* (Sievers *et al.*, 2011) to our new multi-objective dynamic programming algorithm (*MODP*). The latter is implemented in an extended version supporting affine gap costs. To allow for a comparison and as the multi-objective algorithms *Cofolga2mo* and *MODP* return a set rather than one optimal alignment, we evaluated the performance measures for each solution individually and used their maximum.

We want to measure the ability of the tools to reconstruct the reference sequence alignment, as well as the reference consensus structure. The quality of a solution is evaluated using the sum-of-pairs score (Thompson *et al.*, 1999). It is defined as the number of character pairs that are aligned both in the predicted and in the reference alignment divided by the number of aligned character pairs in the reference. To compare the predicted secondary

structures of the sequence-structure alignments to the reference, we use the Matthews correlation coefficient (MCC) for structures (Havgaard *et al.*, 2007; Matthews, 1975).

$$MCC = \frac{P_t N_t - P_f N_f}{\sqrt{(P_t + P_f)(P_t + N_f)(N_t + P_f)(N_t + N_f)}} \quad (10)$$

Here, P_t is the number of predicted base pairs that are also annotated in the reference alignment, P_f is the number of falsely predicted base pairs, N_f is the number of annotated base pairs that are not predicted and N_t is the number of positions that are unpaired in the prediction and the reference alignment. An MCC value of -1 means that the predicted and the reference secondary structure have no base pair in common, where an MCC value of 1 indicates a perfectly accurate prediction. For the sequence alignment tools *Clustal Omega* and *Cofolga2mo*, a direct computation of the MCC is not possible, as they do not calculate secondary structures. We used *RNAalifold* to calculate a consensus secondary structure for their solutions.

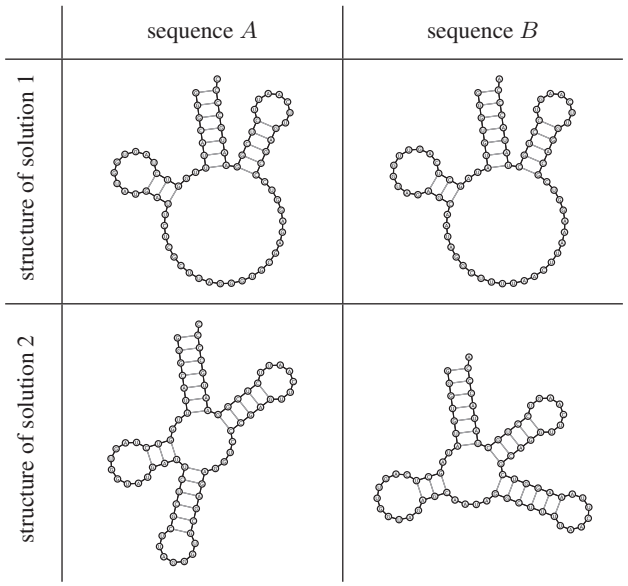

Fig. 5. Secondary structures for the two highlighted solutions of Figure 4. The first row contains the two RNA sequences folded into the structure suggested by Solution 1. The second row contains the more stable structure featuring a fourth stem at the cost of more gaps, as suggested by Solution 2

In Figure 6a, we observe that the sequence alignment tool *Clustal Omega* aligns similar sequences pretty well, but it fails when the sequence identity is <50%. The decreased performance of *P*Mcomp surprises and could be explained by the fact that the weighting parameters in the objective function are not optimal for the dataset used here. *LocARNA*, which has a more sophisticated scoring for the sequence alignment part, shows an increased performance. Both multi-objective algorithms *Cofolga2mo* and *MODP* perform comparably well when the sequence similarity is >40%, but our method achieves good results for dissimilar sequences. As for the MCC score in Figure 6b, which shows the ability of the different methods to predict the reference secondary structure, we see again the expected decreased performance of the purely sequence-based *Clustal Omega* for RNAs with low sequence similarity. The MCC score of *LocARNA* and *PMcomp* both decline as the sequence similarity grows, which could be a consequence of having fixed weighting parameters: if a solution has a high sequence score, the structure score becomes less relevant. Although both multi-objective methods outperform the mono-objective methods, *Cofolga2mo* seems to behave slightly better for sequence pairs with similarity from 30 to 70%. On the borders, however, our approach achieves the higher MCC scores. The evolutionary algorithm *Cofolga2mo* does not explore the solution space completely and possibly misses good solutions, whereas we are able to compute all Pareto-optimal solutions.

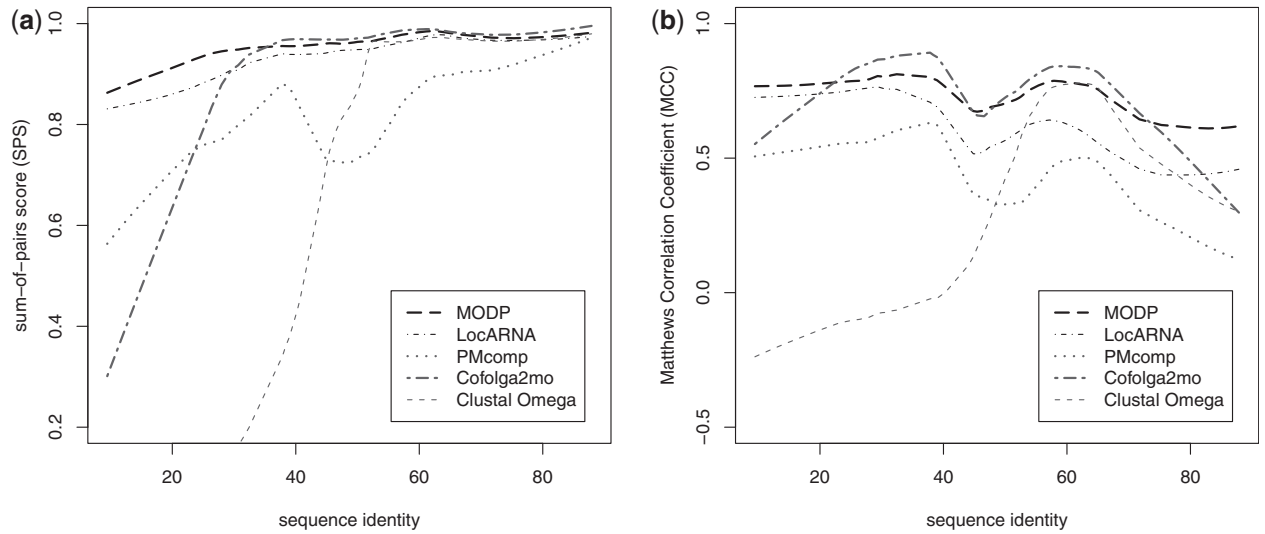

Fig. 6. Sum-of-pairs score (**a**) and Matthews correlation coefficient (**b**) as functions of the sequence identity for dataset 2 of the BRAliBase II benchmark for RNA alignment tools (interpolated using local weighted regression). We compare our method with the tools *LocARNA* and *PMcomp* for RNA sequence-structure alignment and the sequence alignment programs *Cofolga2mo* and *Clustal Omega*, as described in the text

4 DISCUSSION

We introduced a novel method for the construction of pairwise RNA sequence-structure alignments that is inspired by the Sankoff algorithm. To overcome the dependence on fixed weighting parameters for the sequence and structure components in our objective function, we use separate scoring functions for sequence and structure and search for solutions that optimize both objectives. The result is a set of Pareto-optimal solutions from which the human expert is able to choose. This is clearly an advantage to single-objective methods that compute one single solution that is optimal in some sense on which the user has no influence. With the *BRAliBase II* benchmark dataset, we show the effectiveness of our new method compared with well-established tools. Although a hypothetical method that generates all possible solutions would win all benchmarks, albeit being computationally infeasible, all used methods compute in reasonable time. Compared with the evolutionary algorithm of *Cofolga2mo*, our approach has two major features. First, although an evolutionary algorithm is a heuristic method for which statements about the quality of its solutions and the coverage of the solution space are difficult, multi-objective dynamic programming is guaranteed to find all Pareto-optimal solutions to a given problem. Second, we do not only have the ability to produce good sequence alignments but also give all Pareto-optimal solutions to the consensus folding problem of the two RNA sequences. On a practical side, the number of solutions can be limited by clipping the objectives or by using Harrington's (1965) desirability functions, while being able to guarantee finding all solutions in this range. As comparative folding can outperform single-sequence–based methods (Westhof *et al*., 1996), this might be a useful tool for the analysis of novel RNA families. It may even help finding new alternative stable secondary structures for some families of RNA.

All in all, we are able to give a new perspective on the interplay between primary sequence and secondary structure of RNA molecules. These results will also trigger investigations on the potential of multi-objective optimization for the automatic learning of weighting parameters for the established single-objective tools on the basis of published sequence-structure alignments.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers for their very helpful comments that led to significant improvement of this article.

Funding: German Research Foundation (DFG, Scho 302/8-2 to U.S.); Federal Ministry of Education and Research (BMBF, Gerontosys II, Forschungskern SysStaR, project ID 0315894A to H.A.K.).

Conflict of Interest: None declared.

REFERENCES

Backofen,R. *et al*. (2011) Sparse RNA folding: time and space efficient algorithms. *J. Discrete Algorithms*, **9**, 12–31.

Behrouzi,R. *et al*. (2012) Cooperative tertiary interaction network guides RNA folding. *Cell*, **149**, 348–357.

Bellman,R.E. (1957) *Dynamic Programming*. Princeton University Press, New York.

Cleveland,W.S. (1979) Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc*., **74**, 829–836.

Deb,K. (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc, New York, NY.

Doudna,J.A. and Cech,T.R. (2002) The chemical repertoire of natural ribozymes. *Nature*, **418**, 222–228.

Dowell,R. and Eddy,S. (2006) Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*, **7**, 400.

Ehrgott,M. (2000) Multicriteria optimization. *Lecture Notes in Economics and Mathematical Systems*. Vol. 491, Springer-Verlag, Berlin.

Gardner,P.P. *et al*. (2005) A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Res*., **33**, 2433–2439.

Gardner,P.P. *et al*. (2011) Rfam: Wikipedia, clans and the 'decimal' release. *Nucleic Acids Res*., **39**, D141–D145.

Godfrey,P. *et al*. (2007) Algorithms and analyses for maximal vector computation. *VLDB J*., **16**, 5–28.

Gorodkin,J. *et al*. (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res*., **25**, 3724–3732.

Harrington,J. (1965) The desirability function. *Ind. Qual. Control*, **21**, 494–498.

Havgaard,J.H. *et al*. (2007) Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Comput. Biol*., **3**, e193.

Henig,M.I. (1983) Vector-valued dynamic programming. *SIAM J. Control Optimiz*., **21**, 490–499.

Henig,M.I. (1985) The principle of optimality in dynamic programming with returns in partially ordered sets. *Math. Oper. Res*., **10**, 462–470.

Hofacker,I.L. *et al*. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem*., **125**, 167–188.

Hofacker,I.L. *et al*. (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**, 2222–2227.

Klamroth,K. and Wiecek,M.M. (2000) Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Res. Logist*., **47**, 57–76.

Larkin,M.A. *et al*. (2007) Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**, 2947–2948.

Latchman,D. (2005) *Gene Regulation: A Eukaryotic Perspective*. Taylor & Francis, NewYork.

Laux,H. (2005) *Entscheidungstheorie*. 6th edn. Springer-Verlag, Berlin, Germany.

Mathews,D.H. (2005) Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics*, **21**, 2246–2253.

Mathews,D.H. and Turner,D.H. (2002) Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol*., **317**, 191–203.

Matthews,B. (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Pareto,V. (1971) *Manual of Political Economy*. Kelley Publishers, New York, USA.

Roytberg,M. *et al*. (1999) Pareto-optimal alignment of biological sequences. *Biophysics*, **44**, 565–577.

Sankoff,D. (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math*., **45**, 810–825.

Schnattinger,T. *et al*. (2012) Pareto-optimal RNA sequence-structure alignments. In *9th International Workshop on Computational Systems Biology 2012 (WCSB 2012)*, Ulm, Germany, pp 83–86.

Sievers,F. *et al*. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Mol. Syst. Biol*., **7**, 539.

Sniedovich,M. (1988) A multi-objective routing problem revisited. *Eng. Optimiz*., **13**, 99–108.

Taneda,A. (2010) Multi-objective pairwise RNA sequence alignment. *Bioinformatics*, **26**, 2383–2390.

Thompson,J.D. *et al*. (1999) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res*., **27**, 2682–2690.

Tinoco,I. and Bustamante,C. (1999) How RNA folds. *J. Mol. Biol*., **293**, 271–281.

Westhof,E. *et al*. (1996) DNA and RNA structure prediction. In: Bishop,M. and Rawlings,C. (eds) *DNA—Protein Sequence Analysis*. IRL Press, Oxford, pp. 255–278.

Will,S. *et al*. (2007) Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol*., **3**, e65.

Zuker,M. and Sankoff,D. (1984) RNA secondary structures and their prediction. *Bull. Math. Biol*., **46**, 591–621.

Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*., **9**, 133–148.