Preliminary Version for *Advanced Methods* Course ONLY

Incongruences Between Sequence and Secondary Structure Alignments of Nucleic Acids

Maria Waldl[(1)], Christoph Flamm[(1)], Thomas Gatter[(2)], Christian Höner zu Siederdissen[(2)], Michael T. Wolfinger[(1)], Sebastian Will[(1)], Ivo L. Hofacker[(1)], Peter F. Stadler[(2,1,3,4,5)]

[1]Institute for Theoretical Chemistry, Universitä Wien, Währingerstraße 17, A-1090 Vienna., Austria.
{maria,xtof,mtw,ivo,studla,will}@tbi.univie.ac.at
[2]Department of Computer Science and Interdisciplinary Center for Bioinformatics, Universität Leipzig, Härtelstraße 16-18, D-04109 Leipzig, Germany. studla@bioinf.uni-leipzig.de
[3]Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany
[4]Facultad de Ciencias, Universidad National de Colombia, Bogotá, Colombia
[5]Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501, USA

Abstract

This is a preliminary draft of a research manuscript in preparation. It is intended only as accompanying material for the course *"Advanced Methods in Bioinformatics"* at Leipzig University and will soon be superseeded by a complete manuscript.

Motivation:
Independent selection pressures on sequence elements and secondary structure may result in incongruencies between sequence alignments and strueture alignments of the same molecules: Even though the structure are highly similar, base pairs not formed between homologous sequence positions and – conversely – structurally analogous base pairs involve non-homologous sequence positions. No single alignment is capable of faithfully representing such cases. Instead a pair of coupled alignments, one representing sequence evolution and one describing evolution of the structures must be employed.

1 Introduction

Most proteins and RNAs require specific three-dimensional functions to perform their biological functions. As a consequence, mutations tend to preserve secondary structure elements. In RNA this usually implies the conservation of individual base pairs. Families of homologous RNAs therefore often feature a well-defined consensus structure. Nevertheless, there are also abundant variations, in particular insertions and deletions of local structural elements (Brown & Pace, 1991; R. *et al.*, 1994; Williams & Bartel, 1996).

Compensatory substitutions, i.e., substitutions that remedy an earlier (sligthly) deleterious substitution seem to be abundant in both proteins and RNAs (Ivankov *et al.*, 2014). In RNAs, they most commonly result in the replacement of a base pair by another one (Stephan, 1996). In proteins, similar local compensation can be observed albeit it follows less obvious rules (DeJuan *et al.*, 2013). Compensatory evolution is not restricted to substitutions. Although indels are often associated with structural shifts, they may be compensated by indels at other locations(Zhang *et al.*, 2011). For example, the deletion of an aminoacid at the N-terminus of a polypeptide might be compensated by an insertion at the C-terminus. Non-local structural compensation also has been observed for RNAs, where it appears for instance in the form of length compensation of two helices (Lacroix-Labonté *et al.*, 2012). For RNA secondary structure it straightforward to design examples of structural compensation:

```
CGUGGAAACCCACAG   CGUGAAACCUCACAG
.((((....))))..   .((((....))))..
                                      (1)
CGUGGAAACC-CACAG  .((((....))))..
CGU-GAAACCUCACAG  .((((....))))..
```

Instead of preserving the structure exactly, we might also tolerate a shift of the helix

```
CGUGGAAACCCACAG   CGUGGGAAACCCCAG
.((((....))))..   ..((((....)))).
                                      (2)
CGU-GGAAACCCACAG  -.((((....))))..
CGUGGGAAACCC-CAG  ..((((....)))).-
```

The key point about these examples is that some of the analogous base pairs in the two helices are formed by non-homologous bases, and conversely, the evolutionary correct alignment no longer forms a consensus structure. It is, in fact, not difficult to construct shifted structures form very similar sequences even without the need to resort to obvious insertions and deletions:

```
..(((((.((((....)))).))..)))).....
AAGGCUCUAUUAACUGGUAUCGGCUAUAG
** * ***** ***** **** * * ***
AAUGAUCUAUGAACUGUUAUCUGAUUUAG
...(((((.((((....)))).))..)))).....
```
(3)

In this rather extreme example, none of the eight base pairs is preserved even though the secondary is identical except for the shift by a single base pairs and the sequence similarity is just above 75%. Helix shifts are in fact evolutionarily possible even in the absence of indels are possible: As shown by Flamm *et al.* (2000), for any pair of arbitrarily chosen secondary structures of the same length there are sequences that can form both structures, and it is often possible to find sequences for two structural alternatives are dominating the equilibrium ensemble (Höner zu Siederdissen *et al.*, 2013). In summary, negative selection acting to preserve even of large secondary structure elements does not necessarily imply that base pairs are preserved between homolgous nucleotides. Instead, the evolution of sequence and structure may be *incongruent*.

In this contribution we aim to develop a computational method to detect this kind of incongruent evolution in RNAs. We will start from the observation that the alignments implied by sequence similarity and the alignments implies by similarity of secondary structures cannot be reconciled in the form a sequence alignment annotated by a consensus structure. Instead, we formalize the reconciliation of two alignments of the same pair of RNAs as a bi-alignment (Waldl *et al.*, 2019), that is, and alignment of alignments that scores their incongruence. In section 2 we introduce the sequence version of the bi-alignment and show that it reduces to a 4-way alignment problem. When then demonstrate in section 3 that an extension of the Sankoff algorithm for the simultaneous folding and aligning of two RNAs (Sankoff, 1985) solves the bi-alignment problem that models the incongruent evolution of RNA sequences and RNA secondary structures.

2 Bi-Alignments of sequence pairs

Since corresponding structural features are not necessarily formed by homologous sequence position, and homologous sequence positions not necessarily from consensus base pairs, sequence and structure must be represented by separate alignments. The two alignments are not viewed as independent, however. The key idea of *bi-alignments* is to consider them as linked in the following sense: Locally they are congruent if they have the same gap pattern. A difference in the gap patterns of "corresponding" alignment columns, on the other hand, implies an incongruence that *shifts* the two alignments, and thus sequence and structure, relative to each other (Waldl *et al.*, 2019). For illustrative purposes consider the following pair of pairwise alignments:

```
1 2 3 4 5 6 7 8 9      1 2 3 4 - 5 6 7 8 9
m m d d m m d d m      d m m m i m d d d d
a b - - c d - - e      - a b c d e - - - -
```
(4)

where the middle row represents *(mis)matches* `m`, *insertions* `i` and *deletions* `d`. These strings can also be aligned e.g. as

```
- m m d d m m d d m
  | |     |   | |
d m m m i m d d d d
```
(5)

thus establishing a correspondence between columns of the two alignments. Matches, |, denote congruent steps, while mismatches and indels are incongruent steps. This suggests to formalize the problem as follows:

We consider two sequences $\mathbf{a}$ and $\mathbf{b}$ (in more generality, two finite ordered sets of *positions*) and two alignments of these sequences, denoted by $\mathbb{U}$ and $\mathbb{V}$. The alignments are evaluated with respective scoring functions u and v, for instance a sequence and a structure scoring function. Note that each of these two pairwise alignment is faithfully represented by the string denoting insertions, deletions, and (mis)matches. A pairwise alignment $\mathbb{W}$ of $\mathbb{U}$ and $\mathbb{V}$ is therefore well defined as a pairwise alignment of these strings. Introducing a scoring function w for $\mathbb{W}$, we can describe the Bi-Alignment Problem as finding alignments $\mathbb{U}$, $\mathbb{V}$, and $\mathbb{W}$ ($\mathbb{W}$ being an alignment of $\mathbb{U}$ and $\mathbb{V}$) such that the total score

$$u(\mathbb{U}) + v(\mathbb{V}) + w(\mathbb{W}) \tag{6}$$

is optimized. Before we proceed, we note that the bi-alignment has a natural representation as a 4-way alignment. This is an immediate consequence of the fact that arbitrary 4-way alignments can be obtained by progressively aligning two pairwise alignments, see (Berkemer *et al.*, 2018) for a formal analysis of the compositional properties of alignments over general data structures. In the 4-way alignment both x and y appear twice: every column of $\mathbb{W}$ contains either a column of of both $\mathbb{U}$ and $\mathbb{V}$ or gap in either $\mathbb{U}$ or $\mathbb{V}$, which in the 4-way picture is either a pair of gaps in the first or second pair of rows:

```
- 1 2 3 4 5 6 7 8 9
- m m d d m m d d m
- a b - - c d - - e      - 1 2 3 4 5 6 7 8 9
    | |     | | | |      - a b - - c d - - e
1 2 3 4 - 5 6 7 8 9      1 2 3 4 - 5 6 7 8 9
d m m m i m d d d d      - a b c d e - - - -
- a b c d e - - - -
```
(7)

If the scoring functions u, v, and w are *additive*, i.e. if they are additively composed of scores defined on single alignment columns, then the total score is also additive. Each column in the 4-way alignment contributes to components of the total score from Equ. (6): to $u(\mathbb{U})$ for the first pair of entries unless both are gaps; to $v(\mathbb{V})$ for the second pair of entries unless both are gaps; and to $w(\mathbb{W})$ to score the combination of mis-matches, indels, or gap-pairs between formed by the first two and the last two entries in a column. The column score thus depends only the gap pattern in the column itself.

For additive cost functions, the bi-alignment problem therefore reduces to a 4-way alignment problem with a linear gap cost model (Waldl *et al.*, 2019), which is readily solved by dynamic programming (Lipman *et al.*, 1989). This is not the case, of course, if u or v are scoring functions that are non-local, i.e., for scoring functions consider more than a single alignment column. We note that the bi-alignment problem for sequences my be of interest e.g. a means of studying alternative alignments for different scoring models (Vingron & Waterman, 1994). This is even of interest for RNAs, where one can use a nucleotide scoring model and a position based secondary structure score as used e.g. in (Dalli *et al.*, 2006).

In order to simplify the presentation below we will represent the recursions in grammar form. For pairwise alignments with linear gap costs, for instance, we can write the Needleman-Wunsch recursion as

$$A \to A\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right) \mid A\left(\begin{smallmatrix}\bullet\\-\end{smallmatrix}\right) \mid A\left(\begin{smallmatrix}-\\\bullet\end{smallmatrix}\right) \mid \varepsilon\,, \tag{8}$$

where ε denotes the empty alignment and $\mathtt{m} = \left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)$, $\mathtt{d} = \left(\begin{smallmatrix}\bullet\\-\end{smallmatrix}\right)$ and $\mathtt{i} = \left(\begin{smallmatrix}-\\\bullet\end{smallmatrix}\right)$ denote an alignment column containing a (mis)match, deletion, and insertion, respectively. A bullet $\bullet$ represent a letter and $-$

is used as gap symbol. Note that the alternative cases in Equ. (**??**) are independent of the actual sequences. Only the *scoring* associated with each production depends on the annotation associated with the sequence positions (Giegerich *et al.*, 2004). 4-way alignments can be written in exactly same way, using fifteen 4-dimensional terminals in addition the symbol ε, which now represents with empty 4-way alignment:

$$A \to A\left(\begin{smallmatrix}\bullet\\\bullet\\\bullet\\\bullet\end{smallmatrix}\right) \mid A\left(\begin{smallmatrix}\bullet\\\bullet\\\bullet\\-\end{smallmatrix}\right) \mid A\left(\begin{smallmatrix}\bullet\\\bullet\\-\\\bullet\end{smallmatrix}\right) \mid \cdots \mid A\left(\begin{smallmatrix}-\\\bullet\\\bullet\\-\end{smallmatrix}\right) \mid A\left(\begin{smallmatrix}\bullet\\-\\-\\-\end{smallmatrix}\right) \mid \varepsilon\,. \quad (9)$$

We denote the set of the 15 terminal vectors by $\mathcal{C}$. The scoring function w is evaluated in each colum by comparison of the first and last pair of entries of a 4-way column. Since every difference corresponds to a shift, there is penalty of Δ is the first and third or second and fourth entry are equal, and a penalty of 2Δ if both $\mathbf{a}$ and $\mathbf{b}$ behave differently. This yields the scoring table

	$\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}\bullet\\-\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}-\\\bullet\end{smallmatrix}\right)$	$\left(\begin{smallmatrix}-\\-\end{smallmatrix}\right)$
$\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)$	0	Δ	Δ	2Δ
$\left(\begin{smallmatrix}\bullet\\-\end{smallmatrix}\right)$	Δ	0	2Δ	Δ
$\left(\begin{smallmatrix}-\\\bullet\end{smallmatrix}\right)$	Δ	2Δ	0	Δ
$\left(\begin{smallmatrix}-\\-\end{smallmatrix}\right)$	2Δ	Δ	Δ	-

(10)

Since matching $\left(\begin{smallmatrix}-\\-\end{smallmatrix}\right)$ to $\left(\begin{smallmatrix}-\\-\end{smallmatrix}\right)$ does not occur in valid alignments, we leave it undefined in this table. In order to link grammar and recursion we interpret a each of the terminal vectors also as vector with entries 1 (for a bullet •) and 0 (for a gap character −). Let x be a 4-dimension index vector with entries $x_1, x_3 \in [0, \mathrm{len}(\mathbf{a})]$ and $x_2, x_4 \in [0, \mathrm{len}(\mathbf{b})]$ and let $M(x)$ denote the optimal score of an alignment of the prefixes $\mathbf{a}[1, x_1]$, $\mathbf{b}[1, x_2]$, $\mathbf{a}[1, x_3]$, and $\mathbf{b}[1, x_4]$, where prefixes of the form $[1, 0]$ are considered empty. The entries of the dynamic programming table M satisfy

$$M(x) = \max_{c \in \mathcal{C}} M(x - c) + s(x, c) \quad (11)$$

with base case $M(0) = 0$ and column scores $s(x, c)$. This compact notation for the recursion was introduced by Setubal & Meidanis (1997), see also (Retzlaff & Stadler, 2018). After defining $x_{\mathbb{U}} = \left(\begin{smallmatrix}x_1\\x_2\end{smallmatrix}\right)$, $c_{\mathbb{U}} = \left(\begin{smallmatrix}c_1\\c_2\end{smallmatrix}\right)$, $x_{\mathbb{V}} = \left(\begin{smallmatrix}x_3\\x_4\end{smallmatrix}\right)$, $c_{\mathbb{V}} = \left(\begin{smallmatrix}c_3\\c_4\end{smallmatrix}\right)$, the scoring term can be written as

$$s(x, c) = u_s(x_{\mathbb{U}}, c_{\mathbb{U}}) + v_s(x_{\mathbb{V}}, c_{\mathbb{V}}) + w_s(x, c) \quad (12)$$

such that $u_s(x_{\mathbb{U}}, c_{\mathbb{U}})$ yields, depending on $c_{\mathbb{U}}$ being $\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}\bullet\\-\end{smallmatrix}\right)$, or $\left(\begin{smallmatrix}-\\\bullet\end{smallmatrix}\right)$, the (mis)match, deletion, and insertion scores at the respective positions x_1 and x_2 at $\mathbf{a}$ and $\mathbf{b}$ for the first alignment $\mathbb{U}$; $v_s(x_{\mathbb{V}}, c_{\mathbb{V}})$ yields the analogous scores for $\mathbb{V}$; and the shift scores $w_s(x, c)$ are defined in Equ. (10) in a position-independent manner. They could be made position-dependent without algorithmic changes, hence we write them in full generality here. In the following we will further abbreviate Equ. (9) as $A \to Ac \mid \varepsilon$ with the understanding that c denotes all 15 alternative terminal vectors $c \in \mathcal{C}$.

3 Sankoff-style Bi-Alignments

The Simultaneous Alignment and Folding Problem asks for a pairwise sequence alignment $\mathbb{U}$ of two input sequences $\mathbf{a}$ and $\mathbf{b}$ and a consensus structure φ, i.e., a set of base pairs defined on the columns of $\mathbb{U}$ that optimizes a scoring function that evaluates both the quality of the sequence alignment and the common secondary structures. Depending on the scoring model, different variants of the Sankoff algorithm solve this optimization problem by dynamic programming (Sankoff, 1985). In one class of approaches, exemplified by `dynalign` (Harmanci *et al.*, 2007) and `foldalign` (Havgaard *et al.*, 2007), the full, "loop-based" Turner energy model (Turner & Mathews, 2010) to score the consensus structure. In `pmcomp` (Hofacker *et al.*, 2004) and its successors such as `locarna` (Will *et al.*, 2007), individual base pairs and unpaired positions are scored in an additive manner. In our condensed grammar notation, thinking of both non-terminals and terminals as 2-dimensional, i.e., alignment columns formed from the two input sequences, the underlying recursions are of the form

$$A \to Ac \;\big|\; A\bar{c}A\bar{c} \;\big|\; \varepsilon \quad (13)$$

where $A \to Ac$ describes alignment columns that are unpaired in in the consensus structure φ. These are scored as before describes a sequence alignment. The second term, $A \to A\bar{c}A\bar{c}$ describes the formation of a consensus base pair. It is important to note that the corresponding scoring function contains a term that simultaneously scores the pair of terminals $\bar{c}$, $\bar{c}$, i.e., base pairs.

In order to study bi-alignments composed of pure sequence alignment and an alignment with consensus structure we first introduce a slightly more general optimization problem that is easy to specify and has an efficient exact solution by dynamic programming. We will then show that the mixed sequence/structure bi-alignment problem that we are interested in constitutes a natural restriction of the general case and can be solved with the same algorithmic approach.

Given a multiple alignment $\mathbb{X}$, a consensus structure φ is an annotation of the columns of $\mathbb{X}$ that defines a column either to be unpaired or to belong to a unique base pair. In this contribution we are only interested in *nested structures*, although various classes of non-nested, i.e., pseudoknotted structures might also be of interest. A sub-alignment of $\mathbb{Y}$ of $\mathbb{X}$ is obtained by retaining a subset of the row of $\mathbb{X}$ and removing columns in which only gap characters $(-)$ are retained. The restriction of $\varphi_{\mathbb{Y}}$ of φ to $\mathbb{Y}$ consist of all base pairs of φ between alignment columns that are retained in $\mathbb{Y}$. In other words, if one of the two colums forming a base pair in φ is removed in $\mathbb{Y}$, the other one is relabeled as unpaired in $\varphi_{\mathbb{Y}}$.

Definition 1. *The* Simultaneous Bi-Aligmment and Folding Problem *for two sequences* $\mathbf{a}$ *and* $\mathbf{b}$ *asks for a bi-alignment* $\mathbb{S} :\cong (\mathbb{U}, \mathbb{V}, \mathbb{W})$ *of* $\mathbf{a}$ *and* $\mathbf{b}$ *and a consensus structure* φ *defined on* $\mathbb{S}$*, i.e., a set of disjoint pairs of columns of* $\mathbb{S}$ *that optimizes a score function of the form*

$$\mathrm{score}(\mathbb{S}, \varphi) = u(\mathbb{U}, \varphi_{\mathbb{U}}) + v(\mathbb{V}, \varphi_{\mathbb{V}}) + w(\mathbb{W})$$

where $\varphi_{\mathbb{U}}$ *and* $\varphi_{\mathbb{V}}$ *are the restrictions of* φ *to two constituent sequence alignments. The problem is* additive *if both* u *and* v *are sums of contributions of the columns corresponding to unpaired position and base pairs in* $\varphi_{\mathbb{U}}$ *and* $\varphi_{\mathbb{V}}$*, respectively, and* w *is the column-wise shift score defined above.*

In the additive case, $\mathrm{score}(\mathbb{S}, \varphi)$ is the sum of independent contributions for the unpaired columns and paired column pairs of $(\mathbb{S}, \varphi)$, respectively. More precisely, the cost function associated with the first production, i.e., the alignment of unpaired columns of $(\mathbb{S}, \varphi)$ is $s(x, c)$ as defined in Equ. (12) for sequence alignments. For the paired columns of $(\mathbb{S}, \varphi)$ the scoring function is

$$\begin{aligned} r(x, c; y, d) = {} & u_r(x_{\mathbb{U}}, c_{\mathbb{U}}; y_{\mathbb{U}}, d_{\mathbb{U}}) \\ & + v_r(x_{\mathbb{V}}, c_{\mathbb{V}}; y_{\mathbb{V}}, d_{\mathbb{V}}) \\ & + w_s(x, c) + w_s(y, d) \end{aligned} \quad (14)$$

where the shift scores w_s are again given in Equ. (10) and the functions u_r and v_r yield the scores for consensus base pairs in the second productions.

We are now in the position to derive the dynamic programming recursions solving the Additive Simultaneous Bi-Aligmment and Folding Problem. To this end, we denote by $M(x, y)$ the optimal score of a Sankoff bi-alignment of the infixes of $\mathbf{a}[x_1 + 1, y_1]$, $\mathbf{b}[x_2 + 1, y_2]$, $\mathbf{a}[x_3 + 1, y_3]$, and $\mathbf{b}[x_4 + 1, y_4]$, where indices 1 and 2 correspond to the alignment $(\mathbb{U}, \varphi_{\mathbb{U}})$ and the last two coordinates 3 and 4 refer to $(\mathbb{V}, \varphi_{\mathbb{V}})$. It will be

useful to define the set of 'allowed' index combinations $\mathcal{B}^*$ and the set of 'allowed' pairs of gap patterns $\mathcal{C}^*$ that have finite structure score by

$$\begin{aligned}\mathcal{B}^* &:= \{(x,y)|\exists c,d : r(z,c;y,d) > -\infty\}\\ \mathcal{C}^* &:= \{(c,d)|\exists x,y : r(z,c;y,d) > -\infty\}\end{aligned} \tag{15}$$

We observe that a pair $((x,c),(y,d))$ does not appear in the optimal solution if the alternative of replacing the pair by two unpaired columns (x,c) and (y,d) has a strictly better score. Therefore, we can restrict the set of index pairs and gap patterns to those for which the paired scoring is favorable at least sometimes:

$$\begin{aligned}\mathcal{B}^* &:= \{(x,y)|\exists c,d : r(z,c;y,d) > s(z,c)+s(y,d)\}\\ \mathcal{C}^* &:= \{(c,d)|\exists x,y : r(z,c;y,d) > s(z,c)+s(y,d)\}\end{aligned} \tag{16}$$

Theorem 1. *The scoring table $M(,o)f$ the* Additive Simultaneous Bi-Aligmment and Folding Problem *satisfies, for all index vectors $x < y$ the recursion*

$$M(x,y) = \max\begin{cases}\max\limits_{c\in\mathcal{C}} M(x,y-c)+s(y,c)\\ \max\limits_{\substack{(z,y)\in\mathcal{B}^*\\(c,d)\in\mathcal{C}^*}}\begin{pmatrix}M(x,z-c)+M(z,y-d)\\+r(z,c;y,d)\end{pmatrix}\end{cases} \tag{17}$$

with initial conditions $M(x,x) = 0$ for all index vectors x.

Proof. By induction. *Base case.* Empty alignments have an empty consensus structure and correspond to index pairs (x,x). By definition they have score $M(x,x) = 0$. *Induction step.* The the last column of an alignment, indexed by y, has as its consensus structure either an unpaired column or the 3'-side of a base pair. In the latter case there is a column $z < y$, corresponding to the 5'-side of the base pair, such that the restrictions of the alignment to the columns before z and between z and y, respectively, are again alignments with consensus structure because base pairs of the consensus structure do not cross. It therefore suffices for each index pair (x,y) to consider only alignments with consensus structure If y is unpaired, it may have any gap pattern $c \in \mathbb{C}$. For given c, the previous column has index $y-c$, and thus additivity of the scoring function implies that the optimal alignment with an unpaired last column and gap pattern c has score $M(x,y-c)+s(y,c)$. In the paired case columns z and y have gap patterns c and d. Again invoking additive of the scoring function and the fact that base pairs do not cross, we have optimal scores $M(x,z-c)$ and $M(z,y-d)$ for fixed c and d, thus $M(x,y) = M(x,z-c)+M(z,y-d)+r(z,c;y,d)$ provided the base pair can form. This is the case whenever the input sequences satisfy the base pairing rules, and thus $(z,y) \in \mathcal{B}^*$, and the pair of patterns (c,d) statisfies the pertinent restrictions expressed by $(c,d) \in \mathcal{C}^*$. By construction, $M(x,y)$ is the maximum score obtainable form the optimal choice of c in the unpaired case and the optimal choices of pairing partner z of y and, for each of these the best allowed choice of gap patterns c and d. □

Theorem 2. *For shift penalties $\Delta > 0$, the number of shifts n in any optimal bi-alignment $\mathbb{S} \cong (\mathbb{U},\mathbb{V},\mathbb{W})$ is bounded by $n\Delta \le \delta^*(\mathbf{a},\mathbf{b})$ with*

$$\delta^*(\mathbf{a},\mathbf{b}) := \max_{\mathbb{U}} u(\mathbb{U}) + \max_{\mathbb{U}} v(\mathbb{U}) - \max_{\mathbb{U}}[u(\mathbb{U})+v(\mathbb{U})]\,. \tag{18}$$

Proof. Let $\mathbb{S}$ be optimal with score s^*; let n be its number of shifts. It suffices to observe that $s^* + n\Delta \le \max_{\mathbb{U}} u(\mathbb{U}) + \max_{\mathbb{U}} v(\mathbb{U})$ since the score of $\mathbb{S}$ consists of the score of its two constituent alignments, each of which cannot be better then optimizing the alignments separately, minus the shift penalty $n\Delta$. On the other hand, we have $\max_{\mathbb{U}}[u(\mathbb{U})+v(\mathbb{U})] \le s^*$ since two synchronized copies of an alignment $\mathbb{U}$ form a shift-free bi-alignment with score $u(\mathbb{U})+v(\mathbb{U})$. □

The bound shows that sufficiently large Δ completely rule out shifts in the optimal bi-alignment. In this case the bi-alignment reduces to a simple alignment optimizing the sum the cost functions u and v.

In practical applications, we are primarily interested in the case that the first alignment is (scored as) a sequence alignment. To this end we observe that $\bar{c}A\bar{c}$ in the second production can be viewed as the extension of the aligment of two infixes at both ends. In two dimensions, it therefore can produce all possible pairwise alignments. In order to recover the score of a pure sequence alignment it therefore suffices to score the two terminals independently with the scores for a sequence alignment:

$$u_r(x_{\mathbb{U}},c_{\mathbb{U}};y_{\mathbb{U}},d_{\mathbb{U}}) = u_s(x_{\mathbb{U}},c_{\mathbb{U}}) + u_s(y_{\mathbb{U}},d_{\mathbb{U}}) \tag{19}$$

i.e., additive scores the two columns $x_{\mathbb{U}} = \binom{x_1}{x_2}$ and $y_{\mathbb{U}} = \binom{y_1}{y_2}$ according to the corrsponding gap patterns $c_{\mathbb{U}} = \binom{c_1}{c_2}$ and $d_{\mathbb{U}} = \binom{d_1}{d_2}$, respectively. For the second, structural, alignment $(\mathbb{V},\varphi_{\mathbb{V}})$ we require that consensus base pairs in $\varphi_{\mathbb{V}}$ can be cannot involve gaps and is formed by aligning two canonical base pairs. Thus gap patterns must be of the form

$$(c,d) \in \mathcal{C}' := \left\{\begin{pmatrix}-\\-\\\bullet\end{pmatrix},\begin{pmatrix}\bullet\\-\\\bullet\end{pmatrix},\begin{pmatrix}-\\\bullet\\\bullet\end{pmatrix},\begin{pmatrix}\bullet\\\bullet\\\bullet\end{pmatrix}\right\}^2 \tag{20}$$

and thus we assume $\mathcal{C}^* \subseteq \mathcal{C}'$. Furthermore, the allowed index pairs are restricted to

$$\mathcal{B}' = \left\{(x,y)\;\middle|\;\mathbf{a}[x_3]\mathbf{a}[y_3],\mathbf{b}[x_4]\mathbf{b}[y_4]\in\{GC,CG,GU,UG,AU,UA\}\right\}. \tag{21}$$

We assume that non-canonical base pairs result in a 'forbidden' score of $-\infty$, i.e., we have $\mathcal{B}^* \subseteq \mathcal{B}'$. The recursions in Thm. 1 thus remain unchanged.

The scoring model of `pmcomp` (Hofacker *et al.*, 2004) and `locarna` (Will *et al.*, 2007) evaluates consensus base pairs by the combination of a sequence- and a structure-dependent term. The pairing propensity for base pair (i,j) formed by sequence $\mathbf{a}$ is computed as

$$\psi^{\mathbf{a}}(i,j) = \log\frac{p^{\mathbf{a}}_{ij}}{p_0} / \log\frac{1}{p_0} \tag{22}$$

where $p^{\mathbf{a}}_{ij}$ is base pairing probability (of nucleotides i and j in $\mathbf{a}$), and $\psi^{\mathbf{b}}(k,l)$ is determined analogously. The sequence-dependent component is usually of the form $\tau(i,j,k,l) = R(\mathbf{a}[i],\mathbf{a}[j];\mathbf{b}[k],\mathbf{b}[l])$, where the RIBOSUM score $R(\,.\,)$ is determined as a log-odds ratio for the substitution of pairs of paired bases (Klein & Eddy, 2003) is used. We note in passing that the scores used in `locarna` differ from (Klein & Eddy, 2003) due to the choice of a different different background model. We write $\lambda(i,j,k,l) := \psi^{\mathbf{a}}(i,j)+\psi^{\mathbf{b}}(i,j)+\tau(i,j,k,l)$. For the bialignment, we define the "`locarna`" scoring function for unpaired consensus positions by $s(x,c) = u_s(x_{\mathbb{U}},c_{\mathbb{U}})$, i.e., as a pure sequence contribution affecting $\mathbb{U}$ only. For consensus base pairs, we use Eq.(19) for $\mathbb{U}$ and

$$v_r(x_{\mathbb{V}},\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right);y_{\mathbb{V}},\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)) = \lambda(x_3,x_4,y_3,y_4) - u_s(x_{\mathbb{V}},\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)) - u_s(y_{\mathbb{V}},\left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)) \tag{23}$$

for $\mathbb{V}$, where we have used that only $c_{\mathbb{V}} = d_{\mathbb{V}} = \left(\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right)$ may appear in a consensus base pair. By Thm. 2 the solution of the bialignment problem for large penalties Δ satisfies $\mathbb{U} = \mathbb{V}$ and reduces to solving to sequence-structure alignment problem with score $u(\mathbb{U})+v(\mathbb{U})$, which is exactly the scoring function of `locarna` due to our choices of $s(x,c)$, $u_r(x_{\mathbb{U}},(;\,,.\,))$, and $v_r(x_{\mathbb{V}},(;\,,.\,))$.

4 Heuristic Speedups and Space Savings

The bi-alignment recursions of equ.(17) require $\mathcal{O}(n^{12})$ time and $\mathcal{O}(n^8)$, rendering them unusable in practice. Already the complexity of the underlying Sankoff algorithm ($\mathcal{O}(N^6)$ time and $\mathcal{O}(N^4)$ space), is often prohibitive. `locarna`—and, to some extent, also other RNA alignment tools based on Sankoff's model—introduced heuristics for reducing the complexity and achieving a reasonable performance for real world applications. These ideas carry over to the bi-alignment problem.

For naturally evolved RNAs with a functionally relevant, conserved structure we can expect that the number of shifts in the optimal bi-alignment is small. We therefore restrict the entries of $M_{x,y}$ be limiting the total shift $d(x) = |x_1 - x_3| + |x_2 - x_4|$ in each alignment column x by $d(x) \leq \delta_{\text{max}}$. In addition, we can also limit the shift in every alignment between two columns $x < y$ by $d(y - x) \leq \delta_{\text{max}}$. Note that Thm. 2 implies a non-heuristic bound of $\delta_{\text{max}} = \lfloor \delta^*(\mathbf{a}, \mathbf{b})/\Delta \rfloor$. In practise, even small values of δ_{max} are likely to suffice.

The sparsity of the structure space suggest an essential heuristic of `LocARNA`, which is also of use here. Ignoring base pairs with an equilibrium probability below a threshold value p^*, i.e., setting $\psi^{\mathbf{a}}(i,j) = -\infty$ if $p^{\mathbf{a}}_{ij} < p^*$, reduces the number of candidate base pairs from quadratic to linear for each sequence. This prunes $\mathcal{B}^*$ and, consequently, reduces the computational time complexity of the expensive structure match case in Eq. (17) by a quadratic factor; cf. (Will *et al.*, 2007). Also following `LocARNA`, one can achieve an analogous quadratic improvement in space complexity. To this end, we arrange the computation of matrix entries $M(x,y)$ to compute them in rounds of lexicographically decreasing index vectors x ; each round computes the slice $M(x,.)$. For computing the slice, the algorithm requires (for efficiency, constant time) access to two types of entries: matrix entries in the same slice and entries $M(z,y)$ from the structure case. We gain space by reusing the space for the matrix slice $M(x,\ .\ )$ in every round and storing only the entries $M(z,y)$, which are required in the structure case of the recursion, permanently. The space for the latter can be strongly limited by using a sparse data structure, since we require them only for $y, z \in \mathcal{B}^*$, where $\mathcal{B}^*$ is sparse due to the above argument on base pair candidates.

Thirdly, we sparsify the alignment space based on alignment probabilities in a simple sequence alignment of the input sequences (Do *et al.*, 2008). Based on a partition function variant of sequence alignment, this strategy computes the probability that a prefix alignment of $\mathbf{a}[1..i]$ and $\mathbf{b}[1..j]$ is part of an alignment optimizing the sequence component of the score. In our implementation, we compute only the entries of M that pass some hard cutoff probability θ. In this way, using sufficiently small cut-offs, the DP optimization still considers all relevant alignments under the sequence-structure shift alignment score. Limiting the indices in the sequence component based on this heuristic yields another improvement of the time complexity by a linear factor.

Combining the above heuristics yields a Sankoff bi-alignment algorithm that runs in $\mathcal{O}(n^2)$ time and space. By restricting the computations based on the constant δ_{max} (first heuristic), it achieves the same complexity as LocARNA. However, bi-alignment will require significantly more space and time by a factor polynomial in δ_{max}.

References

Berkemer, S., Höner zu Siederdissen, C. & Stadler, P. F. (2018) Alignments as compositional structures. , . submitted; arXiv:1810.07800.

Brown, J. W. & Pace, N. R. (1991) Structure and evolution of ribonuclease P RNA. *Biochimie,* **73**, 689–697.

Dalli, D., Wilm, A., Mainz, I. & Steger, G. (2006) STRAL: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time. *Bioinformatics,* **22**, 1593–1599.

DeJuan, D., Pazos, F. & Valencia, A. (2013) Emerging methods in protein co-evolution. *Nat Rev Genet,* **14**, 249–261.

Do, C. B., Foo, C.-S. & Batzoglou, S. (2008) A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics,* **24** (13), i68–76.

Flamm, C., Hofacker, I. L., Maurer-Stroh, S., Stadler, P. F. & Zehl, M. (2000) Design of multi-stable RNA molecules. *RNA,* **7**, 254–265.

Giegerich, R., Meyer, C. & Steffen, P. (2004) A discipline of dynamic programming over sequence data. *Sci. Computer Prog.,* **51**, 215–263.

Harmanci, A. O., Sharma, G. & Mathews, D. H. (2007) Efficient pairwise RNA structure prediction using probabilistic alignment constraints in `Dynalign`. *BMC Bioinformatics,* **8**, 130.

Havgaard, J. H., Torarinsson, E. & J., G. (2007) Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Comput Biol.,* **3**, 1896–1908.

Hofacker, I. L., Bernhart, S. H. F. & Stadler, P. F. (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics,* **20**, 2222–2227.

Höner zu Siederdissen, C., Hammer, S., Abfalter, I., Hofacker, I. L., Flamm, C. & Stadler, P. F. (2013) Computational design of RNAs with complex energy landscapes. *Biopolymers,* **99**, 1124–1136.

Ivankov, D., Finkelstein, A. & Kondrashov, F. A. (2014) A structural perspective of compensatory evolution. *Curr. Op. Struct. Biol.,* **26**, 104–112.

Klein, R. J. & Eddy, S. R. (2003) Finding homologs of single structured RNA sequences. *BMC Bioinformatics,* **4**, 44.

Lacroix-Labonté, J., Girard, N., Lemieux, S. & Legault, P. (2012) Helix-length compensation studies reveal the adaptability of the VS ribozyme architecture. *Nucleic Acids Res.,* **40**, 2284–2293.

Lipman, D. J., Altschul, S. F. & Kececioglu, J. D. (1989) A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA,* **86**, 4412–4415.

R., G. R., Larsen, N. & Woese, C. R. (1994) Lessons from an evolving ribosomal RNA: 16S and 23S rRNA structure from a comparative perspective. *Microbiological Reviews,* **58**, 10–26.

Retzlaff, N. & Stadler, P. F. (2018) Partially local multi-way alignments. *Math. Comp. Sci.,* **12**, 207–234.

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

Setubal, J. C. & Meidanis, J. (1997) *Introduction to computational molecular biology*. PWS Pub.

Stephan, W. (1996) The rate of compensatory evolution. *Genetics,* **144**, 419–426.

Turner, D. H. & Mathews, D. H. (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucl. Acids Res.,* **38**, D280–D282.

Vingron, M. & Waterman, M. S. (1994) Sequence alignment and penalty choice: review of concepts, case studies and implications. *J. Mol. Biol.,* **235**, 1–12.

Waldl, M., Will, S., Wolfinger, M., L., H. I. & Stadler, P. F. (2019) Bi-alignments as models of incongruent evolution and RNA sequence and structure. In *CIBB*. accepted; BioArxiv: 10.1101/631606.

Will, S., Missal, K., Hofacker, I. L., Stadler, P. F. & Backofen, R. (2007) Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comp. Biol.,* **3**, e65.

Williams, K. P. & Bartel, D. P. (1996) Phylogenetic analysis of tmRNA secondary structure. *RNA,* **2**, 1306–1310.

Zhang, Z., Huang, J., Wang, Z., Wang, L. & Gao, P. (2011) Impact of indels on the flanking regions in structural domains. *Mol. Biol. Evol.,* **28**, 291–301.