Bi-alignments as Models of Incongruent Evolution of RNA Sequence and Secondary Structure

Maria Waldl, Sebastian Will, Peter F. Stadler, Michael T. Wolfinger, Ivo L. Hofacker

Bi-Alignments as Models of Incongruent Evolution of RNA Sequence and Secondary Structure*

Maria Waldl[1 [0000−0001−7098−5712]], Sebastian Will[1 [0000−0002−2376−9205]], Michael T. Wolfinger[1,2 [0000−0003−0925−5205]], Ivo L. Hofacker[1,2 [0000−0001−7132−0800]], and Peter F. Stadler[3,1 [0000−0002−5016−5191]]

[1] University of Vienna, Faculty of Chemistry, Dept. of Theoretical Chemistry, Währingerstraße 17, 1090 Vienna, Austria.
{maria,will,mtw,ivo}@tbi.univie.ac.at

[2] University of Vienna, Faculty of Computer Science, Research Group Bioinformatics and Computational Biology, Währingerstraße 29, 1090 Vienna, Austria.

[3] Dept. of Computer Science and Interdisciplinary Center for Bioinformatics, Leipzig University, Härtelstraße 16-18, 04109 Leipzig, Germany; Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany; Facultad de Ciencias, Universidad National de Colombia, Bogotá, Colombia; Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501, USA studla@tbi.univie.ac.at

Abstract. RNA molecules may be subject to independent selection pressures on sequence and structure. This can, in principle, lead to the preservation of structural features without maintaining the exact position on the conserved sequence. Consequently, structurally analogous base pairs are no longer formed by homologous bases, and homologous nucleotides do not preserve their structural context. In other words, the evolution of sequence and structure is incongruent. We model this phenomenon by introducing bi-alignments, defined as a pair of alignments, one modeling sequence homology; the other, structural homology, together with an alignment of the two alignments that models the relative shifts between conserved sequence and conserved structure. Bi-alignments therefore form a special class of four-way alignments. A preliminary survey of the Rfam database suggests that incongruent evolution is not a very rare phenomenon among structured ncRNAs and RNA elements.

Keywords: RNA secondary structure · RNA alignment · incongruent evolution · 4-way alignment

1 Introduction

The secondary structure of many functional RNAs is well conserved over long evolutionary timescales. Paradigmatic examples include rRNAs, tRNAs, spliceo-

* A preliminary version of this contribution was presented at the 16th International Conference on Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB 2019) [13].

somal RNAs, small nucleolar RNAs, the precursors of miRNAs, many families of regulatory RNAs in bacteria, as well as some regulatory features in mRNAs, such as iron-responsive (IRE) or selenocystein insertion (SECIS) elements. The `Rfam` database [7] collects these RNAs and presents them as an alignment of sequences from different species annotated by a consensus secondary structure. In such families, the variation of the secondary structure is limited to small deviations from the consensus (additional or omitted base pairs). Even more stringently, the notion of a consensus structure implies that conserved base pairs are formed by pairs of homologous nucleotides.

If selection acts to preserve base pairs, then base pairs provide additional information on the homology of nucleotides. As a consequence, Sankoff's algorithm [10] to simultaneously compute an alignment of the sequences and a consensus structure results in an improvement of both the alignment—over structure-unaware sequence alignment—and the predicted secondary structure—over 'homology-unaware' prediction from the single RNAs. Although this assumption of *congruent evolution* of sequence and structure is appealing and has been very fruitful for modeling RNA families in the `Rfam` database, our partial survey of `Rfam` reveals several families that do not follow congruent evolution.

Fig. 1 shows two alignments of the sequence and secondary structure of two paralogous subfamilies of mir-30 precursors. The two families presumably are a product of the vertebrate-specific (2R) genome duplication and have evolved independently for the last 600 Myr. While the two alignments agree in the outer part of the stem loop structure, we observe that the structure alignment (bottom) slightly misaligns well matching sub-sequences, in order to properly align corresponding structure. These substructures are aligned nicely by the sequence alignment (top), which shows a much weaker consensus structure. As key observation, sequence and structure cannot be reconciled in this case. Insisting on matching common sequence patterns necessarily disrupts base pairs, while matching up the base pairs implies that the corresponding sequences appear "shifted" relative to each other.

In this contribution, we introduce a very simple mechanism to go beyond the typical assumption of strong negative selection on both sequence and structure. To bring about incongruencies between sequence and structure, we moreover assume (1) that the selective pressures on sequence and structure are mechanistically independent, and (2) the exact position of the individual base pairs are less important than the overall 'shape' (e.g. the cloverleaf of a tRNA) of the secondary structure. For example, in such a model, a stem may "move" by losing a base pair on one end and introducing a new base pair at the other end. While this kind of stem moving would still be consistent with a consensus structure, in which all inner base pairs are conserved, this consensus structure would not capture the actual analogy of the structure. Even more remarkably, our model allows for more unusual evolutionary transitions.

In the simple example of evolutionary stem sliding (Fig. 2) the sequences of the two sides of a stem or entire stem-loop structure allow two different pairings with disjoint sets of base pairings but comparable energy. Single substitutions

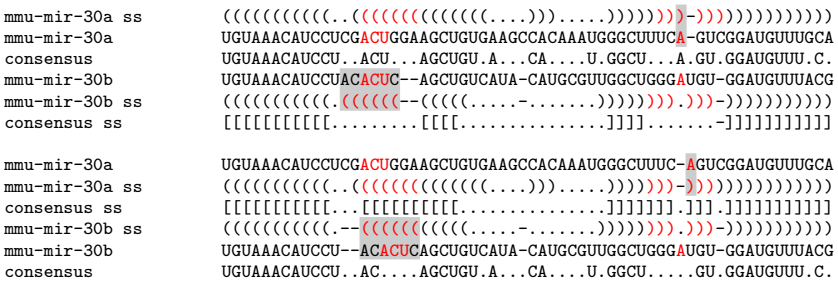

Fig. 1. Two alignments of the mouse miRNA precursors mir-30a and mir-30b, comparing once their sequences (top) and once their mfe structures (bottom). The alignments are annotated by consensus sequences and consensus secondary structures. The two paralogous miRNAs have diverged since the genome duplications in the ancestor of the jawed vertebrates. Aligning the miRNAs based on their sequence (top) suggests that parts of the stem structure have shifted relative to the sequence (shown in red). The identically matched nucleotides in these regions are highlighted in the same way. Due to this shift the highlighted structure parts are misaligned and, therefore, do not appear in the consensus structure. The structure-based alignment (bottom) manages to match the structure almost perfectly, while in turn the underlying sequences are partially misaligned. To visually emphasize this difference between the two alignments, we color the same base pairs and nucleotides as in the upper alignment; moreover, the subsequences that are shifted between the two alignments are marked by gray boxes. The sequence and structure alignment of the two miRNAs cannot be reconciled in a single alignment, but differ by slight shifts. This suggests incongruent evolution of sequence and structure.

or indels may stabilize either one or the other structural alternative, leading to very similar sequences that also have very similar structures, while base pairs are no longer conserved for homologous nucleotides. As a consequence, the sequence alignment (describing *homologous* nucleotides) and the alignment of secondary structures (describing *analogous* base pairs) are incongruent. Stem sliding may explain the evolution of the mir-30 paralogs in Fig. 1: selective pressures at sequence level are dominated by stabilizing selection on the mature miRNA product, while pressures on the structure require only a sufficiently stable stem-loop structure to maintain Dicer processing, independent of the exact position of the mature product in the precursor hairpin.

The incongruence between sequence and structure alignment violates the assumptions underlying the consensus structure model: sequence-based alignments cannot capture the incongruent structure; consequently tools such as `RNAalifold` [2] that determine consensus structures are likely to fail. Conversely, tools like `RNAforester` [6] can produce nearly perfect alignments of the analogous structure. However, due to incongruent evolution, the resulting alignments match non-homologous bases, such that the turnover of base pairs is grossly overpredicted. Usually, combined sequence/structure alignments based on the

```
.(((((....))))).    .(((((....-))))).
ACCCCCUCCGGGGGGA    ACCCCCUCCG-GGGGGA
CCCCCCUCCGGGGGGA    CCCCCCUCCG-GGGGGA
CCCCCCUCCCGGGGGA    -CCCCCCUCCCGGGGGA
(((((.....)))))).   -(((((.....))))).
```

Fig. 2. Evolutionary stem sliding. The two hairpins shown in "dot-parenthesis" notation have no base pair in common. The middle structure folds into both structures with similar energy, the mutants fix different alternatives.

Sankoff algorithm [10], such as `LocARNA` [14] or alternatives such as `cmfinder` [15] would be superior to either sequence-based alignment of structure-based alignment. However, here they do not provide remedy, since these methods rely on congruent evolution of sequence and structure and insist on analogous base pairs being formed by homologous nucleotides.

Incongruent evolution calls for a novel formal framework that enables capturing incongruent evolutionary changes mathematically, which serves as a basis for developing algorithmic approaches to systematically study this phenomenon.

2 Theory

2.1 Bi-Alignments

We embrace the idea that the evolution of sequence and structure of two RNAs **a** and **b** is properly modeled by a pair of alignments $\mathbb{U}$ and $\mathbb{V}$, where $\mathbb{U}$ is intended to represent sequence homology, while $\mathbb{V}$ represents structural similarities. If the sequence and structure evolve congruently, then $\mathbb{U}$ and $\mathbb{V}$ coincide, i.e. both have the same sequence of (mis)match, insertion, and deletion columns, respectively. In the incongruent case, the gap pattern sequences in $\mathbb{U}$ and $\mathbb{V}$ differ. This is captured by an alignment $\mathbb{W}$ of the two alignments $\mathbb{U}$ and $\mathbb{V}$ of **a** and **b**. We call the triple $(\mathbb{U}, \mathbb{V}, \mathbb{W})$ a *bi-alignment* of **a** and **b**. It is a well-known fact that an alignment of a pair of pairwise alignments is a four-way alignment (e.g., see [1] for a formal discussion of the (de)composition of alignments). Thus we can think of a bi-alignment also as a corresponding four-way alignment $\mathbb{A} \simeq (\mathbb{U}, \mathbb{V}, \mathbb{W})$. Each column of $\mathbb{A}$ consists of a column of $\mathbb{U}$ and a colum of $\mathbb{V}$ (match or mismatch column of $\mathbb{W}$), or a column of $\mathbb{U}$ or $\mathbb{V}$ padded by a pair of gaps (insertion or deletion in $\mathbb{W}$), see Fig. 3.

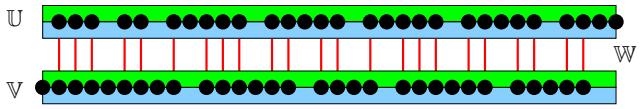

Fig. 3. A bi-alignment consists of two pairwise alignments $\mathbb{U}$ and $\mathbb{V}$ of the same sequences (shown as the two colored bars). Their columns (shown as black balls) in turn arranged in a pairwise alignment $\mathbb{W}$ indicated by the red lines.

A shift between $\mathbb{U}$ and $\mathbb{V}$ occurs whenever an alignment column of $\mathbb{W}$ consumes a different number of sequence positions in **a** or **b** (or both)—thus, changing the lengths of aligned prefixes differently in the respective copies of **a** or **b**. In other words, a shift in a given alignment column of $\mathbb{A}$ corresponds to a difference in the gap patterns of the two copies of **a** (or **b**). We write the gap patterns of pairwise alignments mnemonically as $\binom{\bullet}{\bullet}$ (mismatch), $\binom{\bullet}{-}$ (deletion), $\binom{-}{\bullet}$ (insertion), and $\binom{-}{-}$ (gap column). Algebraically, we interpret them as differences between the prefix length of the current and previous alignment columns, i.e., as $\binom{1}{1}$, $\binom{1}{0}$, $\binom{0}{1}$, and $\binom{0}{0}$, respectively. For an alignment column $\binom{c_{\mathbb{U}}}{c_{\mathbb{V}}}$ we can therefore write the number of shifts as $\|c_{\mathbb{U}} - c_{\mathbb{V}}\|$, i.e., 0 if the gap patterns coincide, 1 if there is a difference for only one **a** or **b**, and 2 if there is difference for each of the sequences. The possible combinations are

	$\binom{\bullet}{\bullet}$	$\binom{\bullet}{-}$	$\binom{-}{\bullet}$	$\binom{-}{-}$
$\binom{\bullet}{\bullet}$	0	1	1	2
$\binom{\bullet}{-}$	1	0	2	1
$\binom{-}{\bullet}$	1	2	0	1
$\binom{-}{-}$	2	1	1	-

(1)

The combination $\binom{-}{-}/\binom{-}{-}$ corresponds to an all-gap column in $\mathbb{A}$ and therefore does not appear. Each column of $\mathbb{A}$ that involves $\binom{-}{-}$ in either $\mathbb{U}$ or $\mathbb{V}$ corresponds to an insertion or deletion in $\mathbb{W}$, otherwise the column is a (mis)match in $\mathbb{W}$. Thus each column of $\mathbb{W}$ is scored by the shift penalty $-\Delta\|c_{\mathbb{U}} - c_{\mathbb{V}}\|$. The sum of the shift penalties over all columns of $\mathbb{W}$ defines the natural score $s_*(\mathbb{W})$ of $\mathbb{W}$. Finally, we define the *bi-alignment problem* as finding a triple $(\mathbb{U}, \mathbb{V}, \mathbb{W})$ that optimizes the sum $s_1(\mathbb{U}) + s_2(\mathbb{V}) + s_*(\mathbb{W})$.

In the four-way alignment $\mathbb{A} \simeq (\mathbb{U}, \mathbb{V}, \mathbb{W})$ we evaluate the two constituent alignments $\mathbb{U}$ and $\mathbb{V}$ (i.e., the first and second pair of rows, respectively), based on (mis)match scores μ, which in practice differ for $\mathbb{U}$ and $\mathbb{V}$ and can be position-specific. Furthermore, gaps in either alignment are linearly penalized based on the gap cost γ. The shift penalty of the column $\binom{c_{\mathbb{U}}}{c_{\mathbb{V}}}$ of $\mathbb{A}$ is $-\Delta\|c_{\mathbb{U}} - c_{\mathbb{V}}\| = -\Delta(|c_1 - c_3| + |c_2 - c_4|)$. Therefore, shift cost in $\mathbb{A}$ can be interpreted as a kind of linear gap cost Δ for indels in the pairwise alignments of the two copies of **a** and **b** (as contained in $\mathbb{A}$). The remaining two pairs of rows 1&4 and 2&3, resp., are not scored at all. In consequence, assuming additive cost models for both $\mathbb{U}$ and $\mathbb{V}$, the components $s_1(\mathbb{U})$, $s_2(\mathbb{V})$, and $s_*(\mathbb{W})$ of the scoring function are additively composed from the column-wise scores, depending on the 15 possible gap patterns

a	•	•	-	•	•	•	-	•	-	•	-	•	-	-	-
b	•	-	•	•	•	-	•	•	•	-	-	-	•	-	-
a	•	•	-	•	-	•	•	-	•	-	•	-	-	•	-
b	•	-	•	-	•	•	•	-	-	•	•	-	-	-	•
$s_1(\mathbb{U})$	μ	γ	γ	μ	μ	γ	γ	μ	γ	γ	0	γ	γ	0	0
$s_2(\mathbb{V})$	μ	γ	γ	γ	γ	μ	μ	0	γ	γ	μ	0	0	γ	γ
$s_*(\mathbb{W})$	0	0	0	Δ	Δ	Δ	Δ	2Δ	2Δ	2Δ	2Δ	Δ	Δ	Δ	Δ

(2)

While in the first three of these 15 cases both alignments move "in sync" and thus incur no shift penalty, all remaining cases induce shifts. In four of them, the alignments move out of sync simultaneously for both input sequences, thus incurring twice the shift penalty.

Let $M(x)$, with $x = (x_1, x_2, x_3, x_4)$ denote the score of the optimal four-way alignment of the prefixes $\mathbf{a}[1..x_1]$, $\mathbf{b}[1..x_2]$, $\mathbf{a}[1..x_3]$, and $\mathbf{b}[1..x_4]$. Depending on the gap pattern $c = (c_1, c_2, c_3, c_4)$ of the last alignment column, the prefix lengths up to the previous column are $x - c$. Following [9,11], M therefore satisfies the recursion

$$M(x) = \max_c M(x - c) + s(x, c) \quad \text{and} \quad M(0) = 0\,, \tag{3}$$

where $s(x, c)$ refers to the scoring function defined in Equ.(2), and c runs over the 15 possible gap patterns (not including the all-gap column). This simple dynamic programming recursion can be evaluated in quartic time and memory.

2.2 Limited shifting and complexity

Large numbers of shifts, i.e., large numbers of indels in the "self-alignments" of **a** and **b**, i.e., row pairs 1 & 3 and 2 & 4, resp., are unlikely to be of interest in practice. Shifts thus can rather be strongly restricted, e.g. to 3 positions. This restriction can be easily realized in the algorithm by limiting index differences, such that one evaluates only a 'band' of the 4-dimension DP matrix. Consequently, one evaluates the recursions in quadratic time (w.r.t. the input length). An alternative would be to employ Carillo-Lipman-style bounds [4] to restrict the computational effort.

It might be tempting to simplify the comparison of $\mathbb{U}$ and $\mathbb{V}$ by using either **a** and **b** as a "reference" and to consider the implied alignment of the two copies of **b** (or **a**, respectively) to assess the shifts. Optimizing over such three-way alignments, however, cannot replace the optimization over the (four-way) bi-alignments. To provide a brief argument, note that the three-way alignments with 'reference' **a** (**b**, analogously) can be represented as—and scored like—bi-alignments without shifts between the copies of **a**. Consequentially maximizing the score over the three-way alignments, yields a lower bound on the bi-alignment score. The existence of optimal bi-alignments that require shifts between the respective copies of a and b, which can be constructed easily for appropriate scoring schemes, shows that this bound is generally not tight.

A further lower bound could be obtained by constructing the bi-alignment progressively—first constructing alignments $\mathbb{U}^*$ and $\mathbb{V}^*$ (optimizing our respective alignment scores u and v) that, second, are aligned by a pairwise alignment optimizing the shift score. Denote with L the minimum of these lower bounds. In contrast, $u(\mathbb{U}^*) + v(\mathbb{V}^*)$ yields an upper bound. Immediately, this allows bounding the number of shifts $\#s$ in optimal alignments by

$$\#s \leq \frac{L - u(\mathbb{U}^*) - v(\mathbb{V}^*)}{\Delta}. \tag{4}$$

It remains open how strongly these bounds improve performance in an Carillo-Lipman approach to bi-alignment.

The (secondary) structure similarity of RNAs typically depends on similarities between corresponding (aligned) base pairs, which introduces a dependency between pairs of alignment columns. Moreover, in many cases the secondary structure of the RNAs is unknown, such that it must be inferred during the alignment. Both issues are addressed by computationally more complex algorithms often following the idea of Sankoff. As algorithmic short cut, one breaks the column dependencies and approximates structure similarity (resembling [3] and `stral` [5]) by a match similarity

$$\mu^S(i,j) = \sqrt{p_1^u(i)p_2^u(j)} + \sqrt{p_1^<(i)p_2^<(j)} + \sqrt{p_1^>(i)p_2^>(j)}, \tag{5}$$

where the $p_k^\bullet(i)$ denote the respective probabilities that position i is unpaired (u), paired upstream ($<$), or downstream ($>$) in the ensemble of RNA $k \in \{1,2\}$).

2.3 Implementation

We implemented our bi-alignment algorithm in Python 3 as a free, open source software tool `BiAlign`; this work describes version 0.2. Our implementation evaluates the structure similarity following Eq. (5). It provides a convenient command line interface and, alternatively, can be integrated as Python module. Both interfaces support full parametrization of the alignment scores and the maximum shift between the sequence copies in the two alignments. Note that setting the maximum shift to zero provides a shift-free base line. Moreover, to facilitate by-eye inspection, the tool highlights conserved sequence and structure and indicates shift events in its output (see Fig. 4).

3 A Survey for Incongruent RNA Evolution

The `Rfam` database provides a large set of curated alignments for structured RNAs. For many of these families the alignments have been created with an emphasis on the consensus structure. In order to identify `Rfam` families in which incongruent evolution may have played a rule, we compare `Rfam 14.1` seed alignments to their `MAFFT` [8] re-alignments. Employing a simple scoring function that considers the sequence positions observed in each column of the alignments, we determined to what extent `Rfam` and `MAFFT` alignments disagree. Incongruent evolution can be ruled out whenever `Rfam` and `MAFFT` alignments are nearly identical. Families for which the alignments disagree strongly, on the other hand, are likely to have experienced incongruent evolution. To limit the computational efforts, we focused on `Rfam` families with small and medium-width seed alignments (≤ 10 sequences, ≤ 120 columns), leaving us with 1181 of 3016 families in `Rfam 14.1`. Out of these we identify 709 cases where the `Rfam` alignment differs from the `MAFFT` realignments.

A more reliable indication for evolutionary shift events is an increase of the combined sequence and structure similarity in the bi-alignment compared to the shift-free baseline. To this end we used our bi-alignment algorithm twice: once to

compute bi-aligments with at most three shifts, and once with forbidden shifts, thus enforcing congruent evolution. We chose *ad hoc*, but plausible parameters for assessing similarity: the sequence similarity in our bi-alignments is simply composed from scoring identical matching nucleotides with 100 (and mismatchs with 0). For assessing structure similarity, we distinguish two cases: for *a priori* unknown structure, we use (mis)match scores defined in Eq. 5; for known structures, we simply count the matched symbols in the dot-bracket structure strings. In order have comparable weights for sequence and structure, structure scores and counts were multiplied by 100. All Indels are scored with -200 and each shift is penalized by -250.

In a second step, we computed the score difference of bi-alignment and shift-free baseline for all 10137 pairs of RNA sequences from the 709 alignments. The optimal bi-alignment exhibits at least one evolutionary shift event in 143 cases from 72 different Rfam families. Fig. 4 shows one example of a very plausible evolutionary shift event. Naturally, the number and significance of predicted shifts strongly depend on the scoring parameters (in particular, shift costs).

4 Multiple Bi-Alignments and Poly-Alignments

The notion of bi-alignments can be generalized to a pair $(\mathbb{U}, \mathbb{V})$ of multiple alignments of the sequences $\mathbf{a^1}, \mathbf{a^2}, \ldots, \mathbf{a^k}$ that are aligned by a pairwise alignment $\mathbb{W}$ (see l.h.s. panel in Fig. 5). Denote the gap pattern in a column of the $\mathbb{W}$-alignment of $\mathbb{U}$ and $\mathbb{V}$ by c and d, resp. The total discrepancy between the gap patterns in the two alignments, i.e., the shift incurred in this column, is $s := \sum_{j=1}^{k} |c_j - d_j|$, where c_j or d_j is the 0-vector (all gaps) for insertions and deletions in $\mathbb{W}$. Note that this coincides with the sum of the number of indels $|c_j - d_j|$ observed in the projected pairwise alignments of the two copies of $\mathbf{a^j}$ with each other.

Let us assume that the k-way alignments $\mathbb{U}$ and $\mathbb{V}$ are scored with a sum-of-pair scores. The corresponding bi-alignment $(\mathbb{U}, \mathbb{V}, \mathbb{W})$ can then be represented as $2k$-way alignment with a scoring function of the form

$$\sigma = s_1(\mathbb{U}) + s_2(\mathbb{V}) + \frac{k-1}{2} s_*(\mathbb{W}) \tag{6}$$

where s_* is again defined by Equ.(2) and the pre-factor $(k-1)/2 = \frac{1}{k}\binom{k}{2}$ accounts for the fact that both $s_1(\mathbb{U})$ and $s_2(\mathbb{V})$ are sums of over $\binom{k}{2}$ pairwise alignments scores, while the shift score is a sum over k contributions. If s_1 and s_2 are additive scoring models, then the $2k$-way alignment also has an overall additive scoring model.

Although the most natural application for bi-alignments is the comparison of sequence and structure, one may want to consider $\ell > 2$ alignments $\mathbb{U}^{(i)}$, $1 \le i \le \ell$ of $\mathbf{a}$ and $\mathbf{b}$ with different scoring schemes. It is then natural to consider a *multiple* alignment $\mathbb{W}$ whose "rows" are the $\mathbb{U}^{(i)}$. We call this a *poly-alignment*, l.h.s. panel in Fig. 5. Writing $\mathbb{W}_{ij}$ for the restriction of $\mathbb{W}$ to the two

`Rfam`:

```
alpaca          ----UGGCUUCCUGCACAUCUGCC--UGUGGCU-AGGGUCACUCCGCGCGGGCACUGUGGCUUUUGUCUGCAUCUGCACAACUAGAUG
alpaca ss       ----.(((.....((((..((((-(((((...-((.....))...)))))))).))))......)))..((((((.....))))))
elephant        GGUGCGGUUUUCCUGUAAGUCGGCCCGUGUGGCUGAGGGUCACUCCGCGCGGGCACUGCAGCUUUUGUCAGCAUCUGGACA-CCUGCCU
elephant ss     ((((.(((.(((((((((.((..((((((((((.(((...))))..)))))))))))))))))))))).(((.....))).....))).)-)))))).
consensus       .....GG.UUCCUG.A..UC.GCCCGUGUGGCU.AGGGUCACUCCGCGCGGGCACUG..GCUUUUGUC.GCAUCUG.ACA.C..G...
consensus ss    .....................[[[.........................]]].................................
```

`BiAlign`:

```
alpaca ss       .(-((---......((--((..(((((((((...-((.....))...)))))))).))))......)))..((((((.....))))))
alpaca          UG-GC---UUCCUGCA--CAUCUGCCUGUGGCU-AGGGUCACUCCGCGCGGGCACUGUGGCUUUUGUCUGCAUCUGCACAACUAGAUG
consensus       .G.GC...UUCCUG.A........C.UGUGGCU.AGGGUCACUCCGCGCGGGCACUG..GCUUUUGUC.GCAUCUG.ACA.CU.G...
elephant        GGUGCGGUUUUCCUGUAAGUCGGCCCGUGUGGCUGAGGGUCACUCCGCGCGGGCACUGCAGCUUUUGUCAGCAUCUGGACACCU-GCCU
elephant ss     ((((.(((.(((((((((.((..((((((((((.(((...))))..)))))))))))))))))))))).(((.....))).....))).)-)))).
consensus ss    ................................................................................

alpaca          UGGCU---UCCUGCAC--AUCUGCCUGUGGCUA-GGGUCACUCCGCGCGGGCACUGUGGCUUUUGUCUGCAUCUGCACAACUA-GAUG
alpaca ss       .(((.---.....(((--(..(((((((((...(-(.....))...)))))))).))))......)))..((((((.....))-))))
consensus ss    .............[[[.....[[[[[[[[.............]]]]]]]]..]]].............................
elephant ss     ((((.(((.(((((((((.((..((((((((((.(((...))))..)))))))))))))))))))))).(((.....))).....))).)-)))).
elephant        GGUGCGGUUUUCCUGUAAGUCGGCCCGUGUGGCUGAGGGUCACUCCGCGCGGGCACUGCAGCUUUUGUCAGCAUCUGGACACCU-GCCU
consensus       .G......U.C............CC....G.....GG......C.....GG.........UUU..............AC..-G...
```

Fig. 4. `Rfam` alignment and bi-alignment of conserved region 1 of the long non-coding RNA Six3os1 (RF02246) from alpaca (ABRR01379223.1/276-356) and elephant (AAGU03061906.1/13962-14048). The alignments are based on the minimum free energy structures of the sequences, which are annotated next to their corresponding sequences (marked with '`ss`'). The consensus sequence string ('`consensus`') indicates conserved nucleotides by capital letters. Matched base pairs appear in the consensus structure strings ('`consensus ss`') as balanced '`[]`' pairs. The minimum free energy structures of both sequences are essentially characterized by a large hairpin. The `Rfam`-based alignment (top) reveals high sequence conservation while the corresponding consensus structure only recovers three base pairs of the putative analogous hairpin structure. In contrast, the bi-alignment by `BiAlign` (below) reconciles the incongruent sequence homology and structure analogy by introducing shifts. Its sequence-based alignment (first part) recovers almost all of the conserved nucleotides found in the `Rfam`-based alignment (the two respective consensus sequences are highlighted in red). At the same time, in its structure-based alignment part, it matches a larger hairpin, which shows up in the consensus structure. Unlike the weak consensus structure of the Rfam alignment, this suggests an incongruently evolved, analogous helix (consensus structures highlighted in red).

"rows" i and j, a natural score is

$$\sigma = \sum_{i=1}^{\ell} s_i(\mathbb{U}^{(i)}) + \frac{2}{\ell - 1} \sum_{i<j} s_*(\mathbb{W}_{ij})\,. \tag{7}$$

The poly-alignment problem then consists in simultaneously optimizing $\mathbb{U}^{(1)}$, $\mathbb{U}^{(2)}$, …, $\mathbb{U}^{(\ell)}$, and $\mathbb{W}$. The normalization factor $2/(\ell-1)$ compensates for the $\binom{\ell}{2}$ shift contributions in relation to the ℓ pairwise alignments. For additive cost functions, the poly-alignment problems can be seen as a 2ℓ-way alignment problem with an additive cost function.

Of course it is also possible to consider poly-alignments consisting of ℓ multiple alignments of k input sequences. In the additive case this becomes a $k\ell$-way alignment problem with a scoring scheme taking into account each multiple align-

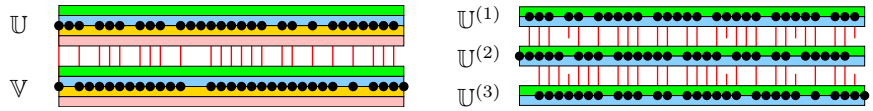

Fig. 5. Generalizations of bi-alignments. L.h.s.: bi-aligment of two multiple (four-way) alignments of the same quadruple of sequences. Bullets denote columns that are present in the constituent alignments, empty spaces indicate insertions/deletions, (i.e. all-gap columns) introduced by the alignment of alignment(s). R.h.s.: poly-alignment comprising three pairwise alignments of the same sequence pair. Red lines indicate the columns of $\mathbb{W}$, the constitutent alignments $\mathbb{U}^{(i)}$ are shown as blocks with a color for each sequence.

ment as well as the indels between the copies of the same sequence. For additive s_i, the poly-alignment again reduces to a $k\ell$-way alignment problem with additive scores. All these generalizations are thus amenable to dynamic programming algorithms and Carillo-Lipman-type sparsification [4] is applicable.

5 Conclusion and Outlook

Incongruent evolution of sequence and structure cannot be captured by the existing RNA alignment methods, which require that consensus structures are formed by homolgous nucleotides. Instead of performing a single common alignment, the sequence and the structure alignment therefore need to be represented separately to account for incongruencies. Bi-alignments appear to be a well-suited mathematical construction for this purpose. Here, we have shown that bi-alignments can be treated as four-way alignments with a scoring function that separately evaluates the two constituent alignments and the shifts between them, i.e., the alignment of the two alignments. The notion of bi-alignments can be generalized naturally to bi-multi-alignments in which both constituent alignment are k-way multiple alignments, and to poly-alignments in which more than two pairwise alignments of the same objects are aligned. In either case one obtains again a multiple alignment with an additive cost model if all constituent alignments use additive costs.

Limiting the total amount of shifts between sequence and structure alignment, the computational cost exceeds the individual alignment problems only by a constant factor. Consequently, bi-alignments are not only of conceptual interest but are also computationally feasible. We provide a Python implementation that can readily be applied to larger data sets.

A survey on a large part of `Rfam` already provides strong indications that incongruent evolution of RNA sequence and secondary structure is not a very rare phenomenon. Shifts of structure relative to sequence seem to have affected at least a few percent of the `Rfam` families. Given that `Rfam` is dedicated to RNA families with well-defined consensus structures, it is plausible to expect that the

phenomenon is even more common in general. It would also be interesting to investigate whether a similar phenomenon can be detected in proteins at the level of secondary structures.

For the conceptual nature of this work, we made two simplifying design choices. Firstly, we chose *ad hoc* parameters for base similarities, indel and shift costs. Although Δ was chosen to penalize shifts more heavily than indels, it would be of great interest to train a more realistic scoring model. Ideally, this could be done in a probabilistic setting as a log-odds ratio, provided a sufficiently large set of shifts can be recovered from empirical data.

Secondly, we follow previous work on sequence-like alignments of structures using scoring models of the form of Equ.(5) [3,5]. In this way, structure-based alignments are modeled as sequence alignments incorporating secondary structures only indirectly via the probabilities of pairing towards the 5'- or 3'-side. This approximation makes the scoring function column-wise additive and thus make it possible to solve the bi-alignment problem by simple recursion (3). Since this model can only approximate the structure-induced dependencies between sequence positions, it does not yield the accuracy of true sequence-structure alignment approaches (e.g. based on Sankoff's algorithm).

We briefly describe the possible extension of our approach to more powerful RNA structure alignment: The (bi-)alignment model presented here corresponds to a regular grammar $A \to Ac|\epsilon$, where c is one of the 15 different possible gap patterns in a four-way alignment [12]. For Sankoff-style structural alignments we additionally need context-free grammar rules of the form $A \to Ac|A(A)|\epsilon$. The alternative production refers to a base pair in the consensus structure. More precisely, this production is of the form $\binom{\mathbb{A}}{\mathbb{B}} \to \binom{\mathbb{A}}{\mathbb{B}}\binom{u}{(}\binom{\mathbb{A}}{\mathbb{B}}\binom{v}{)}$ where the first coordinate refers to the sequence-based alignment and the second coordinate denotes the structural alignment. Here we only allow the insertion of a consensus base pair if both x and y support a base pair at the matching position. In the sequence part we may have $\binom{u}{v} = \binom{\bullet}{\bullet}$, $\binom{\bullet}{-}$, $\binom{-}{\bullet}$, or $\binom{-}{-}$. The four-way version of Sankoff's algorithm has a space complexity of $O(n^8)$ and a time complexity of $O(n^{12})$, which can be reduce to quadratic time and space requirements using the heuristics introduced in `LocARNA` [14] together with a restriction of the shifts. We shall describe Sankoff-style bi-alignments in detail in a forthcoming contribution.

Availability The software package `BiAlign` is freely available at *https://github.com/s-will/BiAlign*

Acknowledgments Partial financial support by the German Federal Ministry of Education and Research (BMBF, project no. 031A538A, de.NBI-RBC and 031L0164C, RNAproNET) and the Austrian science fund (FWF project I 2874 "Prediction of RNA-RNA interactions" and doctoral college W 1207 "RNA Biology") is gratefully acknowledged.

References

1. Berkemer, S., Höner zu Siederdissen, C., Stadler, P.F.: Alignments as compositional structures (2018), submitted; arXiv:1810.07800
2. Bernhart, S.H., Hofacker, I.L., Will, S., Gruber, A.R., Stadler, P.F.: `RNAalifold`: improved consensus structure prediction for RNA alignments. BMC Bioinformatics **9**, 474 (2008)
3. Bonhoeffer, S., McCaskill, J.S., Stadler, P.F., Schuster, P.: RNA multi-structure landscapes. a study based on temperature dependent partition functions. Eur. Biophys. J. **22**, 13–24 (1993)
4. Carrillo, H., Lipman, D.: The multiple sequence alignment problem in biology. SIAM J. Appl. Math. **48**, 1073–1082 (1988)
5. Dalli, D., Wilm, A., Mainz, I., Steger, G.: STRAL: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time. Bioinformatics **22**, 1593–1599 (2006)
6. Hoechsmann, M., Voß, B., Giegerich, R.: Pure multiple RNA secondary structure alignments: A progressive profile approach. IEEE/ACM Trans. Comp. Biol. Bioinf. **1**, 53–62 (2004)
7. Kalvari, I., Argasinska, J., Quinones-Olvera, N., Nawrocki, E.P., Rivas, E., Eddy, S.R., Bateman, A., Finn, R.D., Petrov, A.I.: Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. Nucleic Acids Res. **46**, D335–D342 (2017)
8. Katoh, K., Standley, D.M.: MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol **30**, 772–780 (2013)
9. Lipman, D.J., Altschul, S.F., Kececioglu, J.D.: A tool for multiple sequence alignment. Proc. Natl. Acad. Sci. USA **86**, 4412–4415 (1989)
10. Sankoff, D.: Simultaneous solution of the RNA folding, alignment and protosequence problems. SIAM J. Appl. Math. **45**, 810–825 (1985)
11. Setubal, J.C., Meidanis, J.: Introduction to computational molecular biology. PWS Publications, Boston, MA (1997)
12. Höner zu Siederdissen, C., Hofacker, I.L., Stadler, P.F.: Product grammars for alignment and folding. IEEE/ACM Trans. Comp. Biol. Bioinf. **12**, 507–519 (2015)
13. Waldl, M., Will, S., Wolfinger, M.T., Hofacker, I.L., Stadler, P.F.: Bi-alignments as models of incongruent evolution and RNA sequence and structure. In: CIBB (2019), bioRxiv: 10.1101/631606
14. Will, S., Missal, K., Hofacker, I.L., Stadler, P.F., Backofen, R.: Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. PLoS Comp. Biol. **3**, e65 (2007)
15. Yao, Z., Weinberg, Z., Ruzzo, W.L.: CMfinder–a covariance model based RNA motif finding algorithm. Bioinformatics **22**, 445–452 (2006)