Dynalign II: common secondary structure prediction for RNA homologs with domain insertions

Yinghan Fu1,2, Gaurav Sharma2,3,4,* and David H. Mathews1,2,4,*

1Department of Biochemistry and Biophysics, University of Rochester Medical Center, 601 Elmwood Avenue, Box 712, Rochester, NY 14642, USA, 2Center for RNA Biology, University of Rochester Medical Center, 601 Elmwood Avenue, Box 712, Rochester, NY 14642, USA, 3Department of Electrical and Computer Engineering, University of Rochester, Hopeman 204, RC Box 270126, Rochester, NY 14627, USA and 4Department of Biostatistics and Computational Biology, University of Rochester Medical Center, 601 Elmwood Avenue, Box 630, Rochester, NY 14642, USA

Received July 25, 2014; Revised October 28, 2014; Accepted November 02, 2014

ABSTRACT

Homologous non-coding RNAs frequently exhibit domain insertions, where a branch of secondary structure is inserted in a sequence with respect to its homologs. Dynamic programming algorithms for common secondary structure prediction of multiple RNA homologs, however, do not account for these domain insertions. This paper introduces a novel dynamic programming algorithm methodology that explicitly accounts for the possibility of inserted domains when predicting common RNA secondary structures. The algorithm is implemented as Dynalign II, an update to the Dynalign software package for predicting the common secondary structure of two RNA homologs. This update is accomplished with negligible increase in computational cost. Benchmarks on ncRNA families with domain insertions validate the method. Over base pairs occurring in inserted domains, Dynalign II improves accuracy over Dynalign, attaining 80.8% sensitivity (compared with 14.4% for Dynalign) and 91.4% positive predictive value (PPV) for tRNA; 66.5% sensitivity (compared with 38.9% for Dynalign) and 57.0% PPV for RNase P RNA; and 50.1% sensitivity (compared with 24.3% for Dynalign) and 58.5% PPV for SRP RNA. Compared with Dynalign, Dynalign II also exhibits statistically significant improvements in overall sensitivity and PPV. Dynalign II is available as a component of RNAstructure, which can be downloaded from http://rna.urmc.rochester.edu/RNAstructure.html.

INTRODUCTION

In the past three decades, RNA has been studied not just for its role in protein synthesis, but also for its large number of non-coding roles, where RNA directly controls cellular function (1–6). Because of the biological significance of non-coding RNAs (ncRNAs), the prediction of RNA secondary structure, i.e. the set of canonical base pairs, is now a commonly employed tool for understanding the mechanism of RNA function. Available approaches are categorized and summarized in a number of reviews (7–9).

The most accurate approach for modeling secondary structure is comparative analysis, by which the conserved structure is inferred using multiple homologs. To date, there is no approach that fully automates comparative analysis. One barrier that prevented automation is the fact that folding domains can often be inserted in one homolog relative to another. An inserted domain is a subsequence inserted in one homolog relative to one or more homologs that forms a substructure with base pairing between nucleotides that are within the inserted subsequence. For example, 9.2% of the base pairs in 60 pairs of sequences drawn from a bacterial type A RNase P RNA alignment (10) are in inserted domains. Other barriers include variation of helix and loop length and base pair opening caused by nucleotide mutations between homologous sequences.

This paper describes a novel technique that allows and accounts for domain insertions in prediction of conserved structures for two unaligned sequences. The technique was developed and demonstrated with Dynalign II, an update of Dynalign (11–14), although the principles apply generally to dynamic programming approaches for conserved structure prediction (15–21), including free energy minimization algorithms, partition function algorithms or stochastic context-free grammars. Dynalign is a pairwise RNA secondary structure prediction program that implements the Sankoff algorithm (22) for predicting the conserved structured...
ture for two unaligned homologous sequences; it has also been extended to multiple sequences with the Multilign algorithm (23) and to simultaneous structure prediction with three sequences (24). The dynamic programming recursions were updated in Dynalign II to account for the $\Delta G^o$ in inserted domains. In addition to domain insertions, Dynalign II accommodates other types of structural variations, specifically, base pair openings and stem extensions. Base pair openings represent the situation where one of the homologs has an internal loop with nucleotides that align to base paired nucleotides in the other homolog. Stem extension represent the situation where a helix in one homolog includes a larger number of base pairs than the corresponding helix in the other homolog. The updates to Dynalign handle these structural variations with negligible increase in computational cost by using pre-computed values for the $\Delta G^o$ for inserted domains, obtained from single sequence folding of each homolog.

The developed methodology is validated by benchmarking Dynalign II on ncRNA families that exhibit domain insertions and other structural variations, tRNA, RNase P RNA and SRP RNA. Dynalign II predicts base pairs in inserted domains with better accuracy as compared to Dynalign, and this improvement is statistically significant. Additional tests with SS rRNA homologs provide evidence that Dynalign II encounters no degradation in performance for ncRNA families that have highly conserved secondary structure with little or no structural variation.

The following section highlights the methodology for allowing domain insertions and other aforementioned structural variations. Evaluation methods for benchmarking the algorithm and parameter selection are also discussed within the same section. Next, in the Results section, benchmarks evaluating Dynalign II for accuracy and computation time are presented. The Discussion section closes the paper with concluding remarks and a summary.

**MATERIALS AND METHODS**

**Common secondary structure prediction by $\Delta G^o$ minimization**

Dynalign takes two sequences as input and simultaneously predicts the conserved pseudoknot-free secondary structure and the structural alignment of the sequences. A total $\Delta G^o$:

$$\Delta G_{\text{total}}^o = \Delta G_1^o + \Delta G_2^o + (n_{\text{gap}})\Delta G_{\text{gap penalty}}^o$$  \hspace{1cm} (1)

is minimized, where $\Delta G_1^o$ and $\Delta G_2^o$ are the folding $\Delta G^o$s of sequence 1 and 2, respectively, for the common structure, $\Delta G_{\text{gap penalty}}^o$ is the penalty per gap and $n_{\text{gap}}$ is the number of gaps in the alignment between the two sequences, where the alignment is constrained to be consistent with the common structure. The $\Delta G^o$s are calculated according to the nearest-neighbor thermodynamic model (25–27). While these should technically be referred to as predicted $\Delta G^o$s, the qualifier ‘predicted’ is dropped for brevity. The original Dynalign algorithm (11–14), considers only common structures for which all base pairs in the two homologs are aligned or for which one homolog has single base pairs inserted between two aligned (conserved) base pairs. Therefore, the original Dynalign algorithm does not account for the domain insertions and other structural variations seen in RNA homologs in nature. The same observation holds true for the original Sankoff algorithm and for alternative implementations of the Sankoff algorithm (22). Dynalign II accounts for (the possibility of) domain insertions in one sequence with respect to the other by modifying the total $\Delta G^o$ that is minimized in the process of predicting common structures to

$$\Delta G_{\text{total}}^o = \Delta G_1^o + \Delta G_2^o + (n_{\text{gap}})\Delta G_{\text{gap penalty}}^o + \sum_i (\Delta G_{\text{domain opening}}^o + x_i \Delta G_{\text{domain elongation}}^o)$$  \hspace{1cm} (2)

where $i$ is the index of the $i$th inserted domain, $x_i$ is the length of the $i$th inserted domain and $\Delta G_{\text{domain opening}}^o$ and $\Delta G_{\text{domain elongation}}^o$ are the newly introduced $\Delta G^o$ penalties for initiation and per nucleotide elongation of inserted domains. This is an affine model for each domain insertion into the alignment. $\Delta G_{\text{domain opening}}^o$ and $\Delta G_{\text{domain elongation}}^o$ were optimized on a training data set of known secondary structures (described later) and the value of $\Delta G_{\text{gap penalty}}^o$ was kept the same as in (11). The terms $\Delta G_1^o$ and $\Delta G_2^o$ correspond, as before, to the $\Delta G^o$s of the structures for sequence 1 and sequence 2 according to the nearest-neighbor thermodynamic model.

**Algorithm**

Dynalign II predicts the conserved structure using a dynamic programming algorithm that generalizes the original Dynalign algorithm. In the following discussion, nucleotide positions in each sequence are indexed in 5’ to 3’ order with $i$ and $j$ denoting indices for sequence 1 and $k$ and $l$ denoting indices for sequence 2, with $i < j$ and $k < l$. The optimal structure of a conserved fragment [$i$, $j$, $k$, $l$] of the two input sequences, i.e. the substructure $i$ to $j$ in sequence 1 aligned with the substructure $k$ to $l$ in sequence 2, are determined recursively from smaller to larger fragments by the dynamic programming algorithm. This determines the minimum over all possible pseudoknot-free, common secondary structures and over alignments consistent with those structures. Therefore, the algorithm guarantees the optimal structures will be found given the rules that are implemented. As with Dynalign, Dynalign II predicts structural alignments by only aligning nucleotides that are base paired in conserved base pairs. This is because the $\Delta G^o_{\text{total}}$ in Equations (1) and (2) does not include sequence identity, so nucleotides are not aligned in loop regions.

Given two homologous RNA sequences with lengths $N_1$ and $N_2$, Dynalign fills two 4D arrays of size $N_1 \times N_1 \times N_2 \times N_2$. These arrays are $W(i, j, k, l)$ and $V(i, j, k, l)$, and they represent the $\Delta G^o$ of putative conserved fragments of the two sequences with different conformational constraints. $V(i, j, k, l)$ stores the minimum $\Delta G^o$ of fragments [$i$, $j$, $k$, $l$], where $i$ is base paired with $j$, $k$ is base paired with $l$ and fragment [$i$, $j$] is aligned to fragment [$k$, $l$], $W(i, j, k, l)$ stores the lowest $\Delta G^o$ of fragments [$i$, $j$, $k$, $l$], where fragment [$i$, $j$] is aligned to fragment [$k$, $l$] and these sequence fragments represent potential branches in multibranch loops. In order to fill the arrays, auxiliary 2D arrays are needed, $W3(i, k)$, $W5(i, k)$, $W1_{\text{single}}(i, j)$, $W2_{\text{single}}(k, l)$, $W1_{\text{single}}(i, j)$ and $W2_{\text{single}}(k, l)$, $W3(i, k)$ and $W5(i, k)$ are fragments at the 3’
and 5' end of the two sequences, respectively. \( W_5(i, k) \) stores the minimum \( \Delta G^o \) of fragments \([1, j]\) and \([1, k]\), with no conformational constraints. \( W_3(i, k) \) represents the minimum \( \Delta G^o \) of fragments \([i, N_1]\) and \([k, N_2]\), again with no conformational constraints. \( W_{\text{single}}(i, j), W_{\text{single}}(k, l), W_{\text{single}}(i, j) \) and \( W_{\text{single}}(k, l) \) are newly introduced arrays in Dynalign II for implementing domain insertions. \( W_{\text{single}}(i, j) \) represents the minimum \( \Delta G^o \) of fragment \([i, j]\) of sequence 1 given nucleotides from \( i \) to \( j \) are in a branch in a multibranch loop. \( W_{\text{single}}(k, l) \) is analogously the minimum \( \Delta G^o \) for fragment \([k, l]\) of sequence 2 given that nucleotides from \( k \) to \( l \) are in a branch in a multibranch loop. \( W_{\text{single}}(i, j) \) represents the minimum \( \Delta G^o \) of fragment \([i, j]\), where \( i \) and \( j \) are exterior nucleotides, i.e. there is no base pair \( i'-j' \) where \( i' < i < j < j' \). \( W_{\text{single}}(k, l) \) is for fragment \([k, l]\), and is the analog to \( W_{\text{single}} \) for sequence 2. These four arrays are all calculated using single sequence \( \Delta G^o \) minimization routines in the RNAstructure package (28).

\( V \) for both interior and exterior fragments to facilitate the prediction of suboptimal solutions (12). Interior fragments are those that span nucleotides \( i \) to \( j \) and \( k \) to \( l \). Exterior fragments are those that span nucleotides \( i \) to \( j \) to \( N_1 \), \( 1 \) to \( k \) and \( l \) to \( N_2 \). For conserved structures with base pairs \( i-j \) and \( k-l \), the lowest free energy structure possible is the sum of the \( V \) array for the interior and exterior fragments.

**Overview**

The improvements introduced by the Dynalign II algorithm are illustrated using Figures 1–4 and an abbreviated set of recursions that omit non-essential details. The full set of recursions is available in the Supplementary Materials.

To account for domain insertions, the recursions for \( W(i, j, k, l) \), \( V(i, j, k, l) \), \( W_5(i, k) \) and \( W_3(j, l) \) are modified from the original Dynalign algorithm. \( V(i, j, k, l) \) is determined as

\[
V(i, j, k, l) = \min[V_{\text{hairpin}}(i, j, k, l), V_{\text{internal}}(i, j, k, l), V_{\text{external}}(i, j, k, l), V_{\text{multibranch}}(i, j, k, l) + \text{penalty}(i, j) + \text{penalty}(k, l)]
\]

where \( \text{penalty}(i, j) \) is the penalty term applied to A-U or G-U base pairs at the end of a helix (25,26). \( V_{\text{hairpin}}(i, j, k, l) \) represents the \( \Delta G^o \) of hairpin loops closed by base pairs \( i-j \) and \( k-l \). \( V_{\text{internal}}(i, j, k, l) \) represents the minimum \( \Delta G^o \) of the conserved fragment \([i, j, k, l]\), where internal loops, bulge loops or stacking base pairs are closed by base pairs \( i-j \) and \( k-l \). \( V_{\text{external}}(i, j, k, l) \) accounts for new structural variations incorporated in Dynalign II that include a set of stacking base pairs aligned with an internal loop and insertion of stacking base pairs, internal loops or bulge loops of unlimited length. \( V_{\text{multibranch}}(i, j, k, l) \) represents the minimum \( \Delta G^o \) of the conserved fragment \([i, j, k, l]\), where multibranch loops are closed by base pairs \( i-j \) and \( k-l \). \( V_{\text{domain}}(i, j, k, l) \) represents the minimum \( \Delta G^o \) of the conserved fragment \([i, j, k, l]\), where an inserted domain is formed in a loop closed by base pair \( i-j \) in sequence 1 or \( k-l \) in sequence 2. \( W(i, j, k, l) \) is determined as

\[
W(i, j, k, l) = \min[W_{\text{ext-}}(i, j, k, l), W_{\text{branch}}(i, j, k, l), W_{\text{bifurcation}}(i, j, k, l), W_{\text{domain}}(i, j, k, l)]
\]

**Figure 1.** Expansion of \( W(i, j, k, l) \) to allow domain insertions. (A) and (B) Represent two of the original filling steps of \( W(i, j, k, l) \) that are for conserved domains. (C)–(F) Are expanded steps that allow consideration of inserted domains in four different positions: (C) 3' side of sequence 2, (D) 3' side of sequence 1, (E) 5' side of sequence 2 and (F) 5' side of sequence 1. The black solid lines represent sequences, black dashed lines represent gaps, black arcs represent base pairs and colored brackets are the substructures represented by the array members.

\( W_{\text{ext-}}(i, j, k, l) \) extends substructures shorter by either one or two nucleotides in sequence 1 and/or sequence 2 with unpaired terminal nucleotides. \( W_{\text{branch}}(i, j, k, l) \) considers the formation of a helical branch. \( W_{\text{bifurcation}}(i, j, k, l) \) accounts for bifurcation of \( W(i, j, k, l) \) so that more than three helical branches can be formed in multibranch loops. \( W_{\text{domain}}(i, j, k, l) \) represents the formation of inserted domains to \( W(i, j, k, l) \).

\( W_5(i, k) \) is the minimum \( \Delta G^o \) for substructures from nucleotides \( 1 \) to \( i \) and from \( 1 \) to \( k \). \( W_3(i, k) \) is the minimum \( \Delta G^o \) for substructures \([i, j, k, l]\) and \([1, k]\) of sequence 1 and \([i, k]\) and \([1, l]\) of sequence 2, respectively:

\[
W_5(i, k) = \min[W(i-1, k)+\Delta G_{\text{gap}}, W_5(i-1, k-1)+\Delta G_{\text{gap}}, W_3(i, k-1), W_{\text{branch}}(i, j, k, l), W_{\text{domain}}(i, j, k, l)]
\]
Expansion of $W(i, j, k, l)$ and $V(i, j, k, l)$. In the original Dynalign algorithm, $W(i, j, k, l)$ was the minimum of $W_{\text{extend}}(i, j, k, l)$, $W_{\text{branch}}(i, j, k, l)$ and $W_{\text{bifurcation}}(i, j, k, l)$. The last two terms are given by:

$$ W_{\text{branch}}(i, j, k, l) = V(i, j, k, l) + 2\Delta G^\omega_{\text{helix, terminating in MBL}} \tag{6} $$

$$ W_{\text{bifurcation}}(i, j, k, l) = \min_{i' < j, k' < l} [W(i, i', k, k') + W(i' + 1, j, k' + 1, l)] \tag{7} $$

where Equation (6) represents a single, conserved branch (Figure 1A) with $\Delta G^\omega_{\text{helix, terminating in MBL}}$ being the $\Delta G^\omega$ penalty for terminating a helix of a multibranch loop and Equation (7) represents the bifurcation of the conserved domain (Figure 1B). In order to accommodate domain insertion, the calculation of $W_{\text{domain insertion}}(i, j, k, l)$ is introduced in Dynalign II as:

$$ W_{\text{domain insertion}}(i, j, k, l) = \min [W_{\text{domain insertion1}}(i, j, k, l), W_{\text{domain insertion2}}(i, j, k, l), W_{\text{domain insertion3}}(i, j, k, l), W_{\text{domain insertion4}}(i, j, k, l)] \tag{8} $$

$$ W_{\text{domain insertion1}}(i, j, k, l) = \min_{i' < j} [W(i, i', k, k') + W_{\text{single}}(k' + 1, l) + \Delta G^\omega_{\text{domain opening}} + |l - k'| \Delta G^\omega_{\text{domain elongation}}] \tag{9} $$

$$ W_{\text{domain insertion2}}(i, j, k, l) = \min_{i < j'} [W(i', j', k, k') + W_{\text{single}}(i' + 1, j) + \Delta G^\omega_{\text{domain opening}} + |j' - i'| \Delta G^\omega_{\text{domain elongation}}] \tag{10} $$

$$ W_{\text{domain insertion3}}(i, j, k, l) = \min_{i < i'} [W(i, j, k' + 1, l) + W_{\text{single}}(k, k') + \Delta G^\omega_{\text{domain opening}} + |k' - k| \Delta G^\omega_{\text{domain elongation}}] \tag{11} $$

$$ W_{\text{domain insertion4}}(i, j, k, l) = \min_{i < i'} [W(i, j, k, l) + W_{\text{single}}(i', j) + \Delta G^\omega_{\text{domain opening}} + |j - i'| \Delta G^\omega_{\text{domain elongation}}] \tag{12} $$

where Equations (9)–(12) are illustrated by Figure 1C–F. They represent four possible positions for forming an inserted domain, the 3’ side of sequence 2 ($W_{\text{domain insertion1}}$), the 3’ side of sequence 1 ($W_{\text{domain insertion2}}$), the 5’ side of sequence 1 ($W_{\text{domain insertion3}}$) and the 5’ side of sequence 2 ($W_{\text{domain insertion4}}$). It is important to note that only one variable ($k'$ or $i'$) is enumerated for each equation, and this makes the time scaling $O(N_1^2 + N_2^2)$ for calculating $W_{\text{domain insertion}}(i, j, k, l)$. This is in contrast to $W_{\text{bifurcation}}(i, j, k, l)$, which requires $O(N_1^2 N_2^2)$ time scaling. Therefore, the expansion to account for domain insertion in Dynalign II does not change the time complexity of Dynalign I.

In $V(i, j, k, l)$, $V_{\text{multibranch}}(i, j, k, l)$ is the minimum $\Delta G^\omega$ for pairs closing multibranch loops, i.e.

$$ V_{\text{multibranch}}(i, j, k, l) = \min_{i' < i, j < j', k < k'} [W(i + 1, i', k, k') + W(j + 1, j', k, k') \tag{13} $$

$$ + 2\Delta G^\omega_{\text{helix, terminating in MBL}} + 2\Delta G^\omega_{\text{closure MBL}} ] $$

where two conserved domains form inside base pairs $i$-$j$ and $k$-$l$ (Figure 2A). $\Delta G^\omega_{\text{closure MBL}}$ is the $\Delta G^\omega$ penalty for the closure of a multibranch loop. With just this recursion in $V$ (Equation (3)), a base pair has to close either one conserved domain (forming an internal loop/stacking base pair/bulge loop) or multiple conserved domains (forming a multibranch loop). In order to account for the situation where a conserved base pair closes a different number of branches in one homolog compared to another, the calculation of $V_{\text{domain insertion}}(i, j, k, l)$ is needed:

$$ V_{\text{domain insertion}} = \min [V_{\text{domain insertion1}}(i, j, k, l), V_{\text{domain insertion2}}(i, j, k, l), V_{\text{domain insertion3}}(i, j, k, l), V_{\text{domain insertion4}}(i, j, k, l)] \tag{14} $$

$$ V_{\text{domain insertion1}}(i, j, k, l) = \min_{k < k'} [W(i + 1, j, k + 1, k') + W_{\text{single}}(k' + 1, l - 1) \tag{15} $$

$$ + 2\Delta G^\omega_{\text{helix, terminating in MBL}} + 2\Delta G^\omega_{\text{closure MBL}} + \Delta G^\omega_{\text{domain opening}} + |j - k' - 1| \Delta G^\omega_{\text{domain elongation}} ] $$

$$ V_{\text{domain insertion2}}(i, j, k, l) = \min_{i < j < j'} [W(i, j, k + 1, l - 1) + W_{\text{single}}(i' + 1, j - 1) \tag{16} $$

$$ + 2\Delta G^\omega_{\text{helix, terminating in MBL}} + 2\Delta G^\omega_{\text{closure MBL}} + \Delta G^\omega_{\text{domain opening}} + |j' - i| \Delta G^\omega_{\text{domain elongation}} ] $$

$$ V_{\text{domain insertion3}}(i, j, k, l) = \min_{i < i'} [W(i', j, k + 1, l - 1) + W_{\text{single}}(i, j') \tag{17} $$

$$ + 2\Delta G^\omega_{\text{helix, terminating in MBL}} + 2\Delta G^\omega_{\text{closure MBL}} + \Delta G^\omega_{\text{domain opening}} + |j' - i| \Delta G^\omega_{\text{domain elongation}} ] $$

\[\text{Figure 2.} \] Expansion of $V(i, j, k, l)$ to allow domain insertions. (A) represents the step in the original Dynalign algorithm where two conserved domains form inside a conserved base pair. (B)–(E) Illustrate how the modifications in Dynalign II account for potential inserted domains within the conserved base pair of $V(i, j, k, l)$ at four positions: (B) 5’ side of sequence 2, (C) 3’ side of sequence 1, (D) 5’ side of sequence 2 and (E) the 5’ side of sequence 1.
Equations (15)-(18) are illustrated in Figure 2B–E. By using $W$, $W_1$ and $W_2$ arrays in Equations (14)-(18), any change in the number of branching helices is accommodated because these arrays recursively consider any number of branches (see Equation (7), for example). The form of Equations (15)-(18) allow a multibranch loop in one sequence to structurally align with a single-stem loop rather than a second multibranch loop. In that case, the stem loop would be treated as a branch of a multibranch loop in terms of the energy model. This simplification in the energy model is introduced for computational efficiency.

**Expansion of $W_3(i, k)$ and $W_5(i, k)$.** The two terms in $W_3(i, k)$ and $W_5(i, k)$ arrays exist for adding conserved branches to exterior loops:

$$W_5^\text{bifurcation}(i, k) = \min_{1 \leq s' < i, 1 \leq g' < k} [W_5(i', k') + V(i' + 1, i, k' + 1, k)]$$  \hspace{1cm} (19)

$$W_3^\text{bifurcation}(i, k) = \min_{1 \leq s' < i, 1 \leq g' < k} [W_3(i', k') + V(i', i' - 1, k, k' - 1)]$$  \hspace{1cm} (20)

where Equation (19) is demonstrated in Figure 3A.

Additional terms in the filling of $W_3(i, k)$ and $W_5(i, k)$ arrays are added to consider a domain insertion in exterior loops, $W_5^\text{domain insertion}(i, k)$ and $W_3^\text{domain insertion}(i, k)$:

$$W_5^\text{domain insertion}(i, k) = \min [W_5^\text{domain insertion1}(i, k), W_5^\text{domain insertion2}(i, k)]$$  \hspace{1cm} (21)

$$W_5^\text{domain insertion1}(i, k) = \min_{1 \leq s' < i} [W_5(i', k) + WE_1\text{single}(i' + 1, i) + \Delta G_0^{\text{domain opening}} + |i - i'|\Delta G_0^{\text{domain elongation}}]$$  \hspace{1cm} (22)

$$W_3^\text{domain insertion1}(i, k) = \min [W_3(i', k') + WE_1\text{single}(i', i' - 1) + \Delta G_0^{\text{domain opening}} + |i' - i|\Delta G_0^{\text{domain elongation}}]$$  \hspace{1cm} (23)

Equations (22) and (23) are illustrated in Figure 3B and C.

**Additional structural variations.** In the original Dynalign algorithm, single base pairs could be inserted in one sequence relative to another only if they were flanked by conserved base pairs. In Dynalign II, the model is more flexible. It allows a set of stacking base pairs aligned with an internal loop and unlimited insertion of nucleotides in stacking base pairs, internal loops or bulge loops. In the original Dynalign:

$$V_{\text{internal/stack}}(i, j, k, l) = \min_{1 \leq s < 20.1 \leq l \leq 20.1 \leq d < 20.1} [V(i + a, j - b, k + c, l - d) + \Delta G_{\text{motif}}(i, a, j, b) + \Delta G_{\text{motif}}(k, c, l, d)]$$  \hspace{1cm} (27)

where $\Delta G_{\text{motif}}(m, n, p, q)$ represents the $\Delta G^0$ contributed by a motif, i.e. a base pair stack, internal loop or bulge loop closed by base pairs $m$-$p$ and $n$-$q$ from sequences 1 or 2. In Dynalign II, the additional types of structural alignment are realized (shown in Equations (29)-(32) and Figure 4A–D).
by adding $V_{\text{internal}}/\text{stackII}(i, j, k, l)$:

$$V_{\text{internal}}/\text{stackII}(i, j, k, l) = \min \left\{ V_{\text{internal}}/\text{stackII}(i, j, k, l), V_{\text{internal}}/\text{stackII}(i, j, k, l), V_{\text{internal}}/\text{stackII}(i, j, k, l) \right\}$$

where a set of consecutive base pairs aligned with an internal loop, and $\Delta G_{\text{stack}}(m, m + 1, p, p - 1)$ represents the $\Delta G^o$ contributed by stacking base pair $m-p$ and $(m + 1) - (p - 1)$, which is analogous to $\Delta G_{\text{motif}}(m, m + 1, p, p - 1)$.

$$V_{\text{internal}}/\text{stackII}(i, j, k, l) = \min \left\{ V(i, j, k, c, l - d) + \Delta G_{\text{motif}}(k, k + c, l, l - d), V(i, j, k, c, l - d) + \Delta G_{\text{gap}} \right\}$$

where a motif $k - l$ and $(k + c) - (l - d)$ or $i - j$ and $(i + c) - (j - d)$ is inserted in sequence 2 or 1, respectively, with the gap penalty term added for each unaligned nucleotide.

**Implementation considerations and computational complexity**

The full Dynalign recursions require $O(N_1^3 N_2^3)$ time and $O(N_1^2 N_2^2)$ memory. For typical ncRNA sequence lengths, heuristics for reducing computational time are essential in order to run on current hardware. Dynalign uses an adaptively determined banded constraint on the space of allowable nucleotide alignments. This is based on a hidden Markov model-based estimation of posterior alignment probabilities from the sequences without accounting for structure (13), which requires $O(N_1 N_2)$ time and memory.

If the alignment constraints are approximated by a band with width $d$, i.e. aligned nucleotide indices are no further apart than $(d/2)$, the algorithm reduces to $O(N_1^3 d^2)$ time and $O(N_1^2 d^2)$ memory (22). In addition to the original Dynalign, Dynalign II requires the precomputation of $W1_{\text{single}}(i, j)$, $W2_{\text{single}}(k, l)$, $WE1_{\text{single}}(i, j)$ and $WE2_{\text{single}}(k, l)$, which require $O(N^3)$ computation and $O(N^2)$ memory for each sequence. These are calculated from single sequence secondary structure predictions on each sequence, which are already performed to reduce the set of base pairs considered when filling the $V$ array. This heuristic, which excludes base pairs that can only be found in relatively high $\Delta G^o$ structures, was previously demonstrated to accelerate the calculation with no loss of accuracy (14). Thus, the time and memory complexity of Dynalign II remain the same as Dynalign, despite the additional functionality of handling a greater set of structural variations. Experimental benchmarks presented in the Results section demonstrate that, in agreement with the preceding complexity analysis, the practical time and memory requirements of Dynalign II are also almost identical to those for Dynalign.

**Evaluation**

Two metrics, sensitivity and positive predictive value (PPV), were used to quantify the accuracy of structure predictions for databases of ncRNA families with known secondary structure. Sensitivity is the fraction of known base pairs that are predicted. PPV is the fraction of base pairs predicted that are in the known structure. A predicted base pair $i-j$ is deemed correct if $i-j$, $(i + 1) - j$, $(i - l) - j$, $i - (j - l)$ or $i - (j + l)$ base pair is in the known structure (13,25). This convention is adopted for two important reasons. First, base pairs in RNA structures can be dynamic, for example, single nucleotide bulges can migrate to adjacent nucleotides, as has been observed by nuclear magnetic resonance and by thermodynamic measurements (27,29-30). Second, comparative sequence analysis, which provides the ‘ground-truth’ for evaluating accuracy of secondary structure predictions, is not able to distinguish the two cases encountered when base pairs are able to migrate in position (31). For completeness, metrics computed under an exact matching requirement are also computed and reported in the Supplementary Materials. The absolute difference of all the methods for the four families between exact and flexible matching is 0.031. The maximum difference between exact and the flexible matching is 0.05 and does not change the conclusions for the paper.

For a single sequence pair, sensitivity was calculated as the ratio of the correctly predicted to the total number of known base pairs in the structures of the two sequences, and PPV was computed as the ratio of the correctly predicted to the total number of predicted base pairs in the two sequences. Average sensitivity over an ncRNA family was calculated as the ratio of the correctly predicted to the total number of known base pairs in all the sequence pairs for the family. Average PPV over an ncRNA family was similarly computed as the ratio of the correctly predicted to the total number of predicted base pairs across all the sequence pairs for the family.

Sensitivity and PPV were also computed specifically over base pairs in inserted domains for individual ncRNA families, where complete helices and multibranch loops inserted in one sequence compared to the other homolog in the pair were identified as inserted domains. Here, sensitivity was calculated as the ratio of the correctly predicted to the total number of base pairs in the inserted domains, and PPV was computed as the ratio of the correctly predicted base pairs to the total number of base pairs in the predicted inserted domains.

Because the improvement of accuracy on individual sequence pairs can vary greatly, the one-sided paired t-test procedure of Xu et al. (32) was used to test the null hypothesis that the methods offer identical accuracy against the alternative hypothesis that Dynalign II offers higher accuracy. The one-tail $P$-value was computed to assess statistical significance of the reported improvement in accuracy.
Dynalign II parameters

In addition to the nearest-neighbor thermodynamic parameters, Dynalign II has three additional $\Delta G$ parameters: $\Delta G_{\text{gap penalty}}$, $\Delta G_{\text{domain opening}}$ and $\Delta G_{\text{domain elongation}}$. Among these, $\Delta G_{\text{gap penalty}}$ was determined by maximizing prediction accuracy on 5S rRNA sequences in the original Dynalign (11), and was found to be optimal at 0.4 kcal/mol. $\Delta G_{\text{domain opening}}$ and $\Delta G_{\text{domain elongation}}$ were determined for Dynalign II by a 2D grid search for maximizing prediction accuracy over 66 sequence pairs obtained by selecting all possible pairs from a training data set of 12 group I Intron IC1 subgroup sequences selected from a database of structures (33,34). Based on this procedure, the parameters $\Delta G_{\text{domain opening}}$ and $\Delta G_{\text{domain elongation}}$ were set to 0.5 and 0.1 kcal/mol, respectively. At these chosen values, both sensitivity and PPV were the highest over the training data set. Details of the grid search are provided in the Supplementary Materials.

RESULTS

Structure prediction accuracy was benchmarked using four RNA families: tRNA, RNase P RNA, SRP RNA and 5S rRNA. tRNA sequences can contain variable loops that form inserted stem-loop structures. Forty tRNA sequences were randomly drawn from the Sprinzl database (35) without replacement, and all 780 sequence pairs with these sequences were chosen. The tRNA-inserted base pairs were annotated using tRNAscan-SE 1.21 (36) because the Sprinzl database does not annotate the variable loop base pairs. Base pairs in these inserted domains constitute 2.2% of all base pairs in the sequences. Note that 340 RNase P RNA sequences were randomly drawn without replacement from the bacterial type A RNA alignment on the RNase P database (10) to form 170 non-overlapping sequence pairs. Among all the base pairs in the RNase P RNA data set, 10.7% are in inserted domains. A total of 428 SRP RNA sequence pairs were randomly drawn without replacement from the SRP database (37) to form 214 non-overlapping sequence pairs. Among all the SRP base pairs in the data set, 6.7% are in inserted domains. Twenty 5S rRNA sequence pairs were randomly drawn from the 5S rRNA database (38) without replacement and all 190 possible sequence pairs of these sequences were considered. The 5S rRNA family has no known inserted domains and is included in the benchmark as a test for accuracy of Dynalign II on sequence pairs with little structural variation. Statistics about the pairwise sequence identities for each of the four families are provided as Supplementary Table S7. Four methods were run on the benchmark set: Dynalign II, Dynalign II without domain insertion, (original) Dynalign and Fold (a single sequence $\Delta G$ minimization program from RNAstructure (28)). Dynalign II without domain insertion still included the base pair opening and stem extension functionality in order to separately test the improvement offered by each of the generalizations.

The overall accuracy is illustrated in Figure 5. For the RNase P RNA, SRP RNA and tRNA families, the capability to handle domain insertions and the other two structural variations each improve the sensitivity and PPV. For 5S rRNA, the capability to handle domain insertions does not improve sensitivity or PPV, which is expected given that this family does not have inserted domains. In addition, performance was stratified according to pairwise identity of sequence pairs and the results are reported in Supplementary Table S8.

To further investigate the improvement provided by the capability to account for domain insertions, the accuracy was assessed specifically on base pairs in inserted domains. The results, shown in Figure 6, show that the sensitivity of prediction of base pairs in inserted domains is improved over the original Dynalign algorithm for the RNase P RNA, SRP RNA and tRNA families. Dynalign II also achieves a reasonable PPV in predicting base pairs in inserted domains. Note that the corresponding PPV cannot be calculated for the original Dynalign because inserted pairs are not allowed.

A one-tail paired t-test (32) was performed to test the statistical significance of the improvement in sensitivity and PPV for Dynalign II over Dynalign. The P-values computed for the test are reported in Supplementary Table S9. With the type I error rate, alpha, set to 0.05, the improvements of Dynalign II upon Dynalign are statistically sig-
significant in all cases. The statistical significance of improvements of Dynalign II upon Fold (25) were assessed using the same test and corresponding P-values are included in Supplementary Table S10. All the improvements are significant.

To demonstrate the improvement provided by Dynalign II over Dynalign, an example pair of RNA homologs is illustrated in Figures 7–9. Figure 7 shows the accepted structures for two SRP RNA sequences, *Bacillus amyloliquefaciens* D11416 (SRP database ID: Baci.amyl_D11416) and *Pyrococcus horikoshii* BA000001 (SRP database ID: Pyro.hori_BA000001) (37). *horikoshii* has an inserted domain compared with *amyloliquefaciens* (indicated by a blue rectangle) in addition to the deletion and insertion of base pairs (Figure 7). The prediction made by the original Dynalign algorithm, shown in Figure 8, achieves a sensitivity of 0.55 and a PPV of 0.57. Because the original Dynalign algorithm cannot account for the domain insertion, the overall structures are incorrectly predicted. The prediction from Dynalign II, shown in Figure 9, has an improved sensitivity of 0.86 and PPV of 0.87 (Figure 9). The inserted domain is correctly identified (indicated by a blue rectangle) and the capability to account for the inserted domain also results in an overall more accurate prediction.

The results illustrate the improvement that Dynalign II offers over the original Dynalign in secondary structure prediction accuracy. Advantageously, this improvement is achieved with negligible increase of computational cost. To highlight this, the average run times and memory requirements for the original Dynalign and Dynalign II algorithms are listed in Tables 1 and 2 for RNA sequence pairs from the four families that were used in the accuracy benchmarking. The average execution times and memory requirements for Dynalign II compare favorably with those for Dynalign. For example, the average execution times for the sequence pairs from the RNase P and SRP families were 53 min:41 s and 5 h:6 min:30 s for Dynalign II, compared to 50 min:15 s and 4 h:38 min:37 s for Dynalign, on four cores of an Intel Xeon E5–2695 v2 processor. Similarly, average memory requirements for Dynalign II for sequence pairs from RNase P and SRP families was 812 MB and 1791 MB for Dynalign II, compared to 810 MB and 1790 MB for Dynalign.

**DISCUSSION**

Research aimed at automating comparative sequence analysis has now been ongoing for over a decade. There is still no
algorithm, however, that is as accurate at secondary structure determination as manual effort by an expert investigator (7). Two categories of obstacles prevented this. First, computational methods for comparative sequence analysis fail to properly account for structural variations among homologs. These variations include domain insertions, variations of length of helices, insertions of internal loops/bulge loops and base pair openings caused by mutation of nucleotides. Second, current computational models have only an incomplete comprehension of the factors that impact secondary structure. In particular, the influence of tertiary and pseudoknotted interactions is not included (39,40), and the thermodynamic model is imperfect.

In this paper, a novel methodology was presented to incorporate prediction of inserted domains into dynamic programming algorithms for common secondary structure prediction. The methodology was developed and implemented by updating Dynalign to Dynalign II. Figure 6 shows the dramatic impact that the proposed change has on the ability to correctly predict inserted folding domains. The improvements offered by the new technique over Dynalign in overall average prediction sensitivity and PPV are statistically significant although the numerical gains are small on average because the fraction of base pairs encountered in inserted domains in homologous structures is relatively low (Figure 5). The impact of the proposed change on specific structure predictions can be large, as shown by the example in Figures 7–9, where there is a domain insertion in one sequence relative to the other. Advantageously, the improvement in performance is achieved with negligible increase of computational cost. The new technique generalizes and enhances the overall framework provided by the Sankoff algorithm (22), and is also applicable to other comparative RNA structure analysis tools (7–9). The algorithm presented in this paper accounts for interior inserted domains in multibranch loops that terminate in one or more hairpin stem-loops. Inserted domains, however, can also be found in exterior loops of sequences with known structure, i.e. loops that contain the ends of the sequence, or they can be interior to structures, i.e. not terminating in hairpin stem loops, but terminating in conserved domains.

Another attractive area for further development is to use these improvements for conserved structure prediction for three or more homologous sequences. The work could be extended to multiple sequences, for example, by extending the Multilign method (23) to use Dynalign II instead of Dynalign. Other progressive structure alignment tools could also be adapted in similar ways.

**AVAILABILITY**

Dynalign II is freely available as a component of the RNAstructure package at [http://rna.urmc.rochester.edu/RNAstructure.html](http://rna.urmc.rochester.edu/RNAstructure.html).

**SUPPLEMENTARY DATA**

Supplementary Data are available at NAR Online.

**ACKNOWLEDGEMENT**

The authors thank the Center for Integrated Research Computing, University of Rochester, for providing access to computational resources.

**FUNDING**

National Institutes of Health (NIH) [GM097334 to G.S.]. Funding for open access charge: NIH [GM097334].

**Conflict of interest statement.** None declared.

**REFERENCES**


**Table 1.** Average wall time required for common secondary structure prediction for 5S rRNA, tRNA, RNase P and SRP RNA homologous RNA sequence pairs

<table>
<thead>
<tr>
<th></th>
<th>5S rRNA</th>
<th>tRNA</th>
<th>SRP RNA</th>
<th>RNase P RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynalign II</td>
<td>55 s</td>
<td>18 s</td>
<td>5 h:6 min:30 s</td>
<td>53 min:41 s</td>
</tr>
<tr>
<td>Dynalign</td>
<td>49 s</td>
<td>16 s</td>
<td>4 h:38 min:37 s</td>
<td>50 min:15 s</td>
</tr>
</tbody>
</table>

Four cores of a 12 core Intel Xeon E5–2695 v2 processor (2.4GHz) were used for parallel computations of RNase P RNA and SRP RNA sequence pairs. One core of an Intel Xeon E5–2695 v2 processor (2.4GHz) was used for computations of 5S rRNA and tRNA sequence pairs.

**Table 2.** Average memory required for common secondary structure prediction for 5S rRNA, tRNA, RNase P and SRP homologous RNA sequence pairs

<table>
<thead>
<tr>
<th></th>
<th>5S rRNA</th>
<th>tRNA</th>
<th>SRP RNA</th>
<th>RNase P RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynalign II</td>
<td>76MB</td>
<td>57MB</td>
<td>1791MB</td>
<td>812MB</td>
</tr>
<tr>
<td>Dynalign</td>
<td>74MB</td>
<td>56MB</td>
<td>1790MB</td>
<td>810MB</td>
</tr>
</tbody>
</table>