Predicting Minimum Free Energy Structures of Multi-Stranded Nucleic Acid Complexes Is APX-Hard

Anne Condon
The University of British Columbia, Vancouver, Canada

Monir Hajiaghayi
The University of British Columbia, Vancouver, Canada

Chris Thachuk
The University of Washington, Seattle, WA, USA

Abstract

Given multiple nucleic acid strands, what is the minimum free energy (MFE) secondary structure that they can form? As interacting nucleic acid strands are the basis for DNA computing and molecular programming, e.g., in DNA self-assembly and DNA strand displacement systems, determining the MFE structure is an important step in the design and verification of these systems. Efficient dynamic programming algorithms are well known for predicting the MFE pseudoknot-free secondary structure of a single nucleic acid strand. In contrast, we prove that for a simple energy model, the problem of predicting the MFE pseudoknot-free secondary structure formed from multiple interacting nucleic acid strands is NP-hard and also APX-hard. The latter result implies that there does not exist a polynomial time approximation scheme for this problem, unless P = NP, and it suggests that heuristic methods should be investigated.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Applied computing → Chemistry

Keywords and phrases Nucleic Acid Secondary Structure Prediction, APX-Hardness, NP-Hardness

Digital Object Identifier 10.4230/LIPIcs.DNA.27.9

Funding *Anne Condon*: Supported by an NSERC Discovery Grant.
Monir Hajiaghayi: Supported by an NSERC Discovery Grant.
Chris Thachuk: Supported by a Banting Fellowship, ERC AdG VERIWARE, NSF-CCF-1213127, and NSF-CCF-2106695.

Acknowledgements We thank Erik Winfree for helpful discussions and proposing the problem and we also thank DNA 27 reviewers for their feedback.

1 Introduction

Computational methods are widely used to help understand the structure and function of DNA and RNA molecules. A central challenge has been reliable prediction of nucleic acid secondary structure. In both biological and molecular computing contexts, thermodynamic analyses are widely used for this purpose. Much work has focused on prediction of pseudoknot-free secondary structures, since such structures are common in both biological and designed systems and since pseudoknot-free structures are easier to handle algorithmically [12, 9, 15]. In this paper, we show that, while efficient thermodynamics-based approaches are well known for prediction of pseudoknot-free secondary structures of single strands, the problem of predicting pseudoknot-free secondary structures of multiple interacting strands is computationally intractable unless P = NP. Here and throughout, we consider a method to be efficient if its running time is bounded by a fixed polynomial in the total length of the strands.

In what follows, we briefly summarize significant contributions on development of algorithms for predicting the pseudoknot free secondary structure of a single nucleic acid strand, or of multiple interacting strands. Table 1 also presents a summary of the time complexity of pseudoknot-free secondary structure and partition function prediction. When the input has multiple strands, we separate the cases where the number of strands is bounded by a fixed constant c, and when the number of strands is unbounded, i.e., can grow with the input size. Throughout this work, we focus on the latter case.

Table 1 Computational complexity of predicting nucleic acid MFE pseudoknot-free secondary structures and partition functions, when the input is a single strand, multiple strands with a constant bound c on the number of strands, and multiple strands where the number of strands can grow with the input length n. In each case, n is the total number of bases in the input strand(s). We note that, for a single strand, a work by Bringmann et al. [3] presents an exact sub-cubic algorithm using a simple base pair model. The bold term shows our contribution and the question marks show that the complexity of the corresponding problems is as yet unresolved.

Input Type	MFE	Partition Function
Single Strand	P$[O(n^3)]$ [17, 18, 24]	P$[O(n^3)]$ [16]
Multiple Strands, Bounded ($\leq c$)	?	P$[O(n^3(c-1)!)]$ [4]
Multiple Strands, Unbounded	APX-**hard** [this work]	?

For single strands with length n, dynamic programming algorithms with $O(n^3)$ run time have long been used to efficiently predict minimum free energy (MFE) pseudoknot-free secondary structures, first for a simple "base pair" thermodynamic [17, 18, 24] model in which the free energy of a secondary structure is only dependent on the number of its base pairs, and later for more sophisticated energy models that account for entropic loop penalties, stacked pairs and other structural features. However, very recently, Bringmann et al., [3] proposed a truly sub-cubic algorithm to predict MFE secondary structures for a simple base pair energy model. Dynamic programming methods can also be used to efficiently calculate the partition function for a given strand, making it possible to compute the probability of base pair formation in equilibrium [16].

In addition to prediction of secondary structure of single strands, there has also been much interest in prediction of complexes that result when base pairs form between two or more strands. Such predictions can be used to understand the affinity of binding between a nucleic acid oligonucleotide and its potential target in biological processes such as RNA interference. Prediction of multi-stranded secondary structures is also important because methods for biomolecular programming and construction of nano-devices, such as self-assembly of complex DNA shapes and DNA strand displacement systems, are based on the formation of such complexes [20, 6]; prediction methods such as that provided by NUPACK [22, 5] can guide the design of such programs and devices.

An energy model for single-stranded secondary structure formation can be extended to obtain a model for multi-stranded complex formation by (i) charging an additional strand association penalty, typically a constant times the number of strands involved in the complex, and (ii) accounting for rotational symmetries [4]. Predicting MFE pseudoknot-free secondary structures formed from two (or any constant number) of strands with respect to a model that only accounts for strand association penalties is a straightforward extension of dynamic programming algorithms for single strands [23, 21, 2]. However, it is not clear how such algorithms can efficiently account for rotational symmetries that can arise when two or more indistinguishable strands interact [4]. Nevertheless, Dirks et al. [4] showed how to

efficiently calculate the partition function for a constant number of interacting molecules that form pseudoknot-free structures, by showing how rotational symmetry could be accounted for, while simultaneously addressing algorithmic overcounting issues that arise in partition function calculation. However, the partition function calculation method of Dirks et al. [4] requires a separate dynamic programming computation on all possible orderings of strands that interact to form a single complex. As a result, the method does not run in polynomial time when the number of participating strands grows with the overall input size (total length of strands). This situation can arise, for example, in DNA strand displacement systems. Also, surprisingly, while the partition function for a constant number of interacting strands can be calculated efficiently, it is not known how to efficiently calculate the MFE pseudoknot-free secondary structure of a constant number of interacting strands.

Thus, a basic open question is: can we efficiently compute the MFE pseudoknot-free secondary structure for a multi-set of DNA or RNA strands?

In this paper, we provide a negative answer to this question. Given a set of nucleic acid strands and a positive integer k, let Multi-Pkf-SSP be the problem of determining whether the strands can form a pseudoknot-free secondary structure with at least k base pairs. We show that Multi-Pkf-SSP is NP-hard, meaning that the existence of an efficient method for MFE pseudoknot-free secondary structure prediction of a multi-set of strands would imply all problems in the complexity class NP, which includes problems that are widely believed to be intractable, would have polynomial time algorithms. The hardness result holds whether or not rotational symmetries are accounted for in the energy model. Our proof uses a reduction from a variant of 3-dimensional matching (3DM), already known to be NP-hard, and employs code word designs with high pairwise edit distance [19].

In light of this NP-hardness result, another natural question is whether there is an efficient method to find a pseudoknot-free secondary structure whose energy is a close estimate of the energy of the MFE structure. We also provide a negative answer to this approximation question, by showing a limit to the accuracy of any such method, assuming that $\mathsf{NP} \neq \mathsf{P}$. Specifically, if there is a *polynomial time approximation scheme (PTAS)* that could find a pseudoknot-free secondary structure whose free energy closely approximates that of the MFE for any given multi-set of strands, then again $\mathsf{NP} = \mathsf{P}$. A PTAS is a polynomial time algorithm that receives as input an instance of an optimization problem and an arbitrary parameter $\epsilon > 0$, and returns an output whose value (in our case, the number of base pairs in the MFE structure) is within a factor $1 - \epsilon$ of the value of the optimal solution. The running time of a PTAS could be dependent on ϵ, but it must be polynomial in the input size for every fixed ϵ. Formally, we show that the optimization problem of finding the MFE structure for a multi-set of nucleic acid strands is hard for the complexity class APX, the class of NP optimization problems that have constant factor approximation algorithms. We show this result by establishing that our reduction from 3-dimensional matching to MFE structure prediction is an approximation-preserving reduction.

We note that hardness results have already been proved for variants of pseudoknotted secondary structure prediction. While dynamic programming can be used to predict MFE structures and partition functions for certain restricted classes of pseudoknotted structures, the general problem of predicting MFE pseudoknotted structures is NP-hard, even for a single strand [1, 14, 13]. The first two NP-hardness results, [1, 14] also use a simple energy model called *stacking* where only consecutive base pairs forming a stack contribute to the free energy of a strand. Hardness results can be valuable even with simple energy models; it would seem unlikely that the prediction problem becomes easier if the energy model is more sophisticated.

The rest of the paper is organized as follows. We provide preliminary definitions, problem statements and an overview of some useful theorems in Section 2. We outline the string properties and designs required for our reduction, in Section 3. We provide a polynomial-time reduction from a variant of 3DM to Multi-Pkf-SSP in Section 4, and prove its correctness in Section 5. In Section 6, we also infer that an optimization version of the problem is hard for the complexity class APX. This implies that there is no PTAS for approximating the optimal secondary structure of multi-stranded systems, unless NP = P. The proofs of some lemmas have been moved to Appendix A.

2 Preliminaries

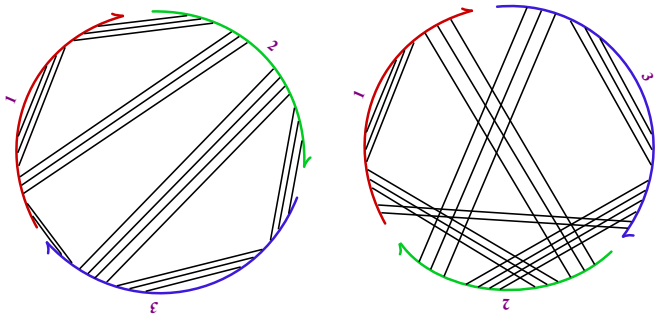
We review some basic terminology and prior work in order to precisely formulate the problem description and proof techniques.

A single DNA or RNA strand is a sequence of nucleotide bases, which we represent using the character set {A, C, G, T} or {A, C, G, U} respectively, with the left end of the sequence corresponding to the 5′ end of the strand and the right end corresponding to the 3′ end. Bonds can form between Watson-Crick base pairs, namely C–G and A–U for RNA and C–G and A–T for DNA [4].

We assume that consecutive bases within a sequence cannot pair with each other. This is consistent with actual structures, where there are typically at least three bases separating any two bases that are paired with each other. If sequences are numbered consecutively starting from 1, we can represent a base pair as a tuple (i, j), such that $i < j - 1$, which specifies that the base at position i in the sequence is paired with the base at position j and j is not consecutive with i. A secondary structure is a set of base pairs such that no base is in two pairs. That is, if (i, j) and (i', j') are in the structure then i, j, i' and j' are all distinct.

Base pairing between two strands occurs in an antiparallel format. That is, the Watson-Crick complement of strand $x = 5'\text{-}x_1 \cdots x_n\text{-}3'$ is the strand $3'\text{-}\bar{x}_1 \cdots \bar{x}_n\text{-}5' \equiv 5'\text{-}\bar{x}_n \cdots \bar{x}_1\text{-}3'$, where $(x_i, \bar{x}_i)$ is a Watson-Crick base pair. For example, the Watson-Crick complement of 5′-ACTCG-3′ is 5′-CGAGT-3′. Throughout we will use the term *complement* to mean Watson-Crick complement and denote the complement of x by $\bar{x}$.

Similar to the single-stranded model, the secondary structure formed by m interacting strands is a set of Watson-Crick base pairs. To specify the secondary structure, we assign identifiers from 1 to m to the strands, and each base is named by a strand identifier and a position on the corresponding strand. For instance if base i in strand s pairs with base j in strand t, where $s \leq t$ and $i < j - 1$ if $s = t$, the base pair is denoted as (i_s, j_t). A multi-stranded secondary structure can be represented as a polymer graph by ordering and depicting the directional (5′ to 3′) strands around the circumference of a circle, with edges along the circumference representing adjacent bases, and straight line edges connecting paired bases. Each such ordering of m strands is a circular permutation of the strands, and there are $(m - 1)!$ possible orderings. A secondary structure consists of one or more *complexes* that correspond to the connected components in the polymer graph representation. If the polymer graph of any one of these possible orderings has no crossing lines, then the secondary structure is called *pseudoknot-free* [4]. For example, Figure 1 shows the two possible circular permutations for three strands 1, 2, and 3, and the connected polymer graphs for the same secondary structure. Since Figure 1(a) has no crossing lines, the structure is pseudoknot-free.



(a) (b)

Figure 1 a) A polymer graph representation of a pseudoknot-free secondary structure for the strand set $\{1, 2, 3\}$ with ordering 123. b) A second polymer graph for the same structure, with strand ordering 132.

2.1 The simple energy model

Here, we employ a very simple extension of the "base pair free energy" model for secondary structures [18]. In that model, the score of each base pair is -1 and the overall score (free energy) of a single-stranded secondary structure is its total number of base pairs. So, the more base pairs in a secondary structure of a single strand, the lower its score.

Where there are multiple interacting strands, there is an entropic penalty for strands to associate via base pairing, i.e., a penalty for reducing the number of complexes [4]. In our simplified model, we define the strand association penalty to be $K_{\text{assoc}} \geq 0$. Thus, for a pseudoknot-free secondary structure S consisting of m strands, l ($\leq m$) complexes, and p base pairs, the overall score, or free energy, of S is

$$\mathbf{E}(S) = p(-1) + (m - l)K_{\text{assoc}}.$$

For example, the secondary structure in Figure 1(a) has score $21(-1) + (3 - 1)K_{\text{assoc}} = -21 + 2K_{\text{assoc}}$. For strands $s_1, \ldots, s_m$, an *optimal* pseudoknot-free secondary structure S_{opt} satisfies $\mathbf{E}(S_{opt}) \leq \mathbf{E}(S)$ for any pseudoknot-free secondary structure S of $s_1, \ldots, s_m$.

Since there can be a tradeoff between the number of base pairs and the number of complexes, then it is possible under this model for an optimal pseudoknot-free secondary structure to have less than the maximum number of possible base pairs. However, our proofs have been constructed so that pseudoknot-free MFE secondary structures will have a maximum number of base pairs for any reasonable value of the constant K_{assoc}. We will proceed with our problem definitions under the assumption that $K_{\text{assoc}} = 0$ and formally argue later that the results hold for all constants $K_{\text{assoc}} \geq 0$.

2.2 Problem definitions

We now formally define the main problem of interest in this paper as a decision problem.

▶ **Problem 1** (Multi-Pkf-SSP).
Instance: m nucleic acid strands and a positive integer k.
Question: Do the m strands form a pseudoknot-free secondary structure containing at least k base pairs?

Figure 2 An instance of the restricted 3-dimensional matching problem, 3DM(3), where $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$, $Z = \{z_1, z_2, z_3\}$. (a) The set of permitted triples, $\mathcal{T} = \{(x_1, y_2, z_2), (x_2, y_1, z_1), (x_2, y_3, z_2), (x_3, y_3, z_3)\}$. (b) A valid matching $\mathcal{M} \subseteq \mathcal{T}$.

We will describe a polynomial-time reduction from a restriction of the NP-hard 3-dimensional matching problem to MULTI-PKF-SSP. A 3-dimensional matching is defined as follows. Let X, Y, and Z be finite, disjoint sets, and let $\mathcal{T}$ be a subset of $X \times Y \times Z$. That is, $\mathcal{T}$ consists of triples (x, y, z) such that $x \in X$, $y \in Y$, and $z \in Z$. Now $\mathcal{M} \subseteq \mathcal{T}$ is a *3-dimensional matching* if the following holds: for any two distinct triples $(x_i, y_j, z_k) \in \mathcal{M}$ and $(x_a, y_b, z_c) \in \mathcal{M}$, we have $x_i \neq x_a$, $y_j \neq y_b$, and $z_k \neq z_c$.

For convenience in our construction, we use a restriction of the 3-dimensional matching problem, called 3DM(3), that requires each element to appear in at most three triples of $\mathcal{T}$.

▶ **Problem 2** (3DM(3)).
Instance: A set $\mathcal{T} \subseteq X \times Y \times Z$, where $|X| = |Y| = |Z| = n$ and each element of X, Y and Z appears in at most 3 triples of $\mathcal{T}$.
Question: Does there exist a matching $M \subseteq T$, with $|M| = n$?

▶ **Theorem 1** (Garey & Johnson (1979) [7]). *3DM(3) is* NP*-complete.*

Next we define an optimization version of the MULTI-PKF-SSP decision problem:

▶ **Problem 3** (MAX-MULTI-PKF-SSP).
Instance: A set of m nucleic acid strands.
Optimization Problem: Determine a pseudoknot-free secondary structure of the m strands with maximum number of base pairs.

An optimization problem is in APX if it has a constant factor approximation algorithm, i.e., an efficient method that can determine a solution whose score is within some fixed multiplicative factor of that of an optimal solution. A problem is APX-hard if for some constant c, a c-approximation algorithm for the problem would imply that NP = P. One way to prove a problem is APX-hard is to show an approximation-preserving reduction from a known APX-hard problem. A problem is APX-complete if it is APX-hard and is in APX. We derive our hardness result for the MAX-MULTI-PKF-SSP problem by a reduction from the MAX-3DM(3) problem, an optimization version of the 3DM(3) problem:

▶ **Problem 4** (MAX-3DM(3)).
Instance: A set $\mathcal{T} \subseteq X \times Y \times Z$, where $|X| = |Y| = |Z| = n$ and each element of X, Y and Z appears in at most 3 triples of $\mathcal{T}$.
Optimization Problem: Find a maximum size 3-dimensional matching $M \subseteq \mathcal{T}$.

Kann [11] showed that MAX-3DM(3) is MaxSNP-complete and thus APX-complete. Hardness of approximation was established by demonstrating that it is NP-hard to decide whether an arbitrary instance of the problem has a matching of size n or a matching of size at most $(1 - \epsilon_0)n$, for some $\epsilon_0 > 0$.

▶ **Theorem 2** (Kann (1994) [11]). *MAX-3DM(3) is APX-complete.*

3 String designs and their properties

In this section we show how to design strings with properties that are useful in our reduction. We follow standard string notation: for a string $a = a_1 \dots a_n$ we denote its i^{th} character (or symbol) by a_i and its length by $|a| = n$; for any symbol B, we let B^l denote a string of length l consisting of only B's. The following related string properties are of particular interest to us.

1. A *pairwise sequence alignment*, or simply *alignment*, of strings a and b is a pair of strings (a', b') with $|a'| = |b'|$, where a' and b' are obtained from a and b respectively by the insertion of zero or more copies of a special *gap* symbol. Moreover, for any i, a'_i and b'_i are not both gap symbols and if neither a'_i nor b'_i is the gap symbol then $a'_i = b'_i$. The alignment can alternatively be considered as a sequence of aligned pairs $(a'_i, b'_i), 1 \leq i \leq |a'|$. A pair is a *gap* pair if either a'_i or b'_i is a gap symbol. We also define an *optimal alignment* of a and b as a pairwise alignment of a and b with a minimum number of gap pairs, amongst all possible alignments.
2. A *longest common subsequence* between strings a and b is a longest subsequence common to the two strings. Note that a subsequence of a string results from the deletion of zero or more of its characters. We denote the length of such a subsequence by $\mathbf{LCS}(a, b)$. A longest common subsequence corresponds to an optimal alignment of a and b and $\mathbf{LCS}(a, b)$ is equal to the total number of gap-free pairs of symbols in the alignment.
3. The *insertion-deletion distance* $\mathbf{d_{LCS}}(a, b)$ between strings a and b is the minimum number of insertions and deletions of symbols needed to convert a into b (or equivalently to convert b to a). Equivalently, the insertion-deletion distance between a and b is equal to the number of gap pairs in an optimal alignment of a and b.

The insertion-deletion distance and length of the longest common subsequence of two strings are related by the following known result.

▶ **Theorem 3** ([8]). *Given two strings a and b, where $|a| = n$ and $|b| = n'$, then $\mathbf{d_{LCS}}(a, b) = k$ if and only if $\mathbf{LCS}(a, b) = \frac{(n+n'-k)}{2}$.*

Note that if a and b are equi-length strings, then k is an even number.

In the next theorem, we show how to efficiently construct a "large" set of relatively short, equi-length strings that have high pairwise insertion-deletion distance. The construction employs a greedy codeword design used also in Justesen [10] and Schulman and Zuckerman [19].

▶ **Theorem 4.** *Let $w > 0$ and $\delta > 0$. For any n, a set of at least wn equi-length strands over the alphabet $\{\mathtt{A}, \mathtt{T}\}$, each of length $k \log n$ for some constant k (that depends on w and δ), can be designed in $2^{O(\log n)}$ time, such that the insertion-deletion distance between any pair in the set is at least $\delta \log_2 n$. Moreover, all strands in the set have at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{A}$'s and at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{T}$'s.*

Proof. We construct the desired set using a greedy algorithm that is specified in terms of a quantity $t = \Theta(\log_2 n)$ that we determine in the penultimate paragraph of this proof. From $\{\mathtt{A}, \mathtt{T}\}^t$, first put the two strings $\mathtt{A}^t$ and $\mathtt{T}^t$ in the set. Once $i \geq 2$ strings are in the set, choose any string from $\{\mathtt{A}, \mathtt{T}\}^t$ whose insertion-deletion distance from all i strings already in the set is at least $\delta \log_2 n$, and add it to the set. Continue until no more strings can be chosen with the desired insertion-deletion distance. Finally, remove the strings $\mathtt{A}^t$ and $\mathtt{T}^t$. This algorithm runs in time $2^{O(\log n)}$. The number of strings in $\{\mathtt{A}, \mathtt{T}\}^t$ that have insertion-deletion distance at most $2d$ from a given string s in $\{\mathtt{A}, \mathtt{T}\}^t$ is at most $\binom{t}{d}^2 2^d$ (see proof of Lemma 2 of Schulman and Zukerman [19]). If $d = \lceil \delta \log_2 n/2 \rceil$, then our set has the desired property that the insertion-deletion distance between any pair in the set is at least $\delta \log_2 n$. Furthermore all strings in the set, once $\mathtt{A}^t$ and $\mathtt{T}^t$ are removed, must have at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{A}$'s and at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{T}$'s; otherwise, their insertion-deletion distance from $\mathtt{A}^t$ and $\mathtt{T}^t$, would be less than $\delta \log_2 n$.

The number of strings in the set before removal of $\mathtt{A}^t$ and $\mathtt{T}^t$ is at least $wn + 2$ if we choose t so that

$$2^t / (\binom{t}{d}^2 2^d) \geq 2^{t/2} \geq wn + 2.$$

These inequalities hold if t is a sufficiently large constant times $\log_2 n$. For the first inequality, from Stirling's formula we have that $\binom{t}{d} < (t\,e/d)^d$, and so the inequality holds if $2d \log_2(t\,e/d) + d \leq t/2$. This in turn holds if $t = \eta d$ $(= \eta \lceil \delta \log_2 n/2 \rceil)$ where we choose constant η so that $\eta e \leq 2^{\eta/4 - 1/2}$. For the second inequality, we simply need that $t \geq 2 + 2\log_2 w + 2\log_2 n$.

Finally, since the strings $\mathtt{A}^t$ and $\mathtt{T}^t$ are removed and all other strings have insertion-deletion distance at least $\delta \log_2 n$ from strings $\mathtt{A}^t$ and $\mathtt{T}^t$, all strands in the set have at least $\delta \log_2 n$ $\mathtt{A}$'s and at least $\delta \log_2 n$ $\mathtt{T}$'s. ◀

Our design also makes use of a *padding* function. Let ρ^i denote the padding function that, applied to a string, inserts i $\mathtt{A}$'s (called padded $\mathtt{A}$'s) at the start of, and between every pair of symbols in, the string.

▶ **Definition 5** (padding function ρ^i)**.** *Let $a = a_1 a_2 \ldots a_n$ be a string. Then $\rho^i(a) = \mathtt{A}^i a_1 \mathtt{A}^i a_2 \ldots \mathtt{A}^i a_n$.*

If $\mathbf{d_{LCS}}(a, b) = k$ then $\mathbf{d_{LCS}}(\rho^i(a), \rho^i(b))$ may be less than k. To illustrate why, first consider the function ρ^1, defined as $\rho^1(a_1 a_2 \ldots a_n) = \mathtt{A}^1 a_1 \mathtt{A}^1 a_2 \ldots \mathtt{A}^1 a_n$. If we choose $a =$ $\mathtt{AATATT}$, and $b = \mathtt{TTATAA}$, then $\mathbf{d_{LCS}}(a, b) = 6$ whereas $\mathbf{d_{LCS}}(\rho^1(a), \rho^1(b)) = 4$, as shown in Figure 3. This appears to contradict the assertion in Lemma 2 of Schulman and Zukerman [19] that $\mathbf{d_{LCS}}(\rho^1(a), \rho^1(b)) \geq \mathbf{d_{LCS}}(a, b)$. Adapting this example, if

$$a' = \mathtt{A}^5\mathtt{A}^5\mathtt{T}^5\mathtt{A}^5\mathtt{T}^5\mathtt{T}^5 \text{ and } b' = \mathtt{T}^5\mathtt{T}^5\mathtt{A}^5\mathtt{T}^5\mathtt{A}^5\mathtt{A}^5,$$

then $\mathbf{d_{LCS}}(a', b') = 30$, while $\mathbf{d_{LCS}}(\rho^5(a'), \rho^5(b')) = 20$.

We next show a lower bound on $\mathbf{d_{LCS}}(\rho^i(a), \rho^i(b))$ in terms of $\mathbf{d_{LCS}}(a, b)$.

```
A  ATATT                    A A AAATAAATAT
 TTATA  A                   ATATAAATAAA A
```

(a) (b)

Figure 3 Padding can reduce insertion-deletion distance. (a) The ATA substrings of the two strings of length 6 forms a LCS, leaving a total of six symbols unmatched. (b) When the strings are 1-padded, the leftmost A of the first string and the rightmost A of the second string, plus the padded A's, become part of the LCS.

▶ **Lemma 6.** *Let a and b be equi-length strings over $\{\mathtt{A},\mathtt{T}\}$. Then*

$$\mathbf{d_{LCS}}(\rho^i(a),\rho^i(b)) \geq \mathbf{d_{LCS}}(a,b)/2.$$

Let a and b be strands and let $S(a)$ and $S(a,b)$ be secondary structures for strand a and pair (a,b) respectively. The base pairs of (a,b) may be inter-molecular and/or intra-molecular. We define the *unpairedness* of $S(a)$ or $S(a,b)$ to be the number of bases that are not paired in $S(a)$ or $S(a,b)$, respectively. The next lemma provides lower bounds on the unpairedness of structures formed from padded strings.

▶ **Lemma 7.** *Let a' and b' be any strands over the alphabet $\{\mathtt{A},\mathtt{T}\}$, let $a = \rho^5(a')$, let $b = \rho^5(b')$, and let s be any substrand of a or $\bar{a}$. Let $S(s)$, $S(a,b)$, $S(\bar{a},\bar{b})$ and $S(a,\bar{b})$ be any pseudoknot-free secondary structures for s, (a,b), $(\bar{a},\bar{b})$ and $(a,\bar{b})$, respectively. Then*

1. *The unpairedness of $S(s)$ is at least $\frac{1}{3}|s|$.*
2. *The unpairedness of $S(a,b)$ is at least $\frac{2}{3}(|a|+|b|)$.*
3. *The unpairedness of $S(\bar{a},\bar{b})$ is at least $\frac{2}{3}(|\bar{a}|+|\bar{b}|)$.*
4. *The unpairedness of $S(a,\bar{b})$ is at least $\frac{1}{3}\mathbf{d_{LCS}}(a,b)$.*

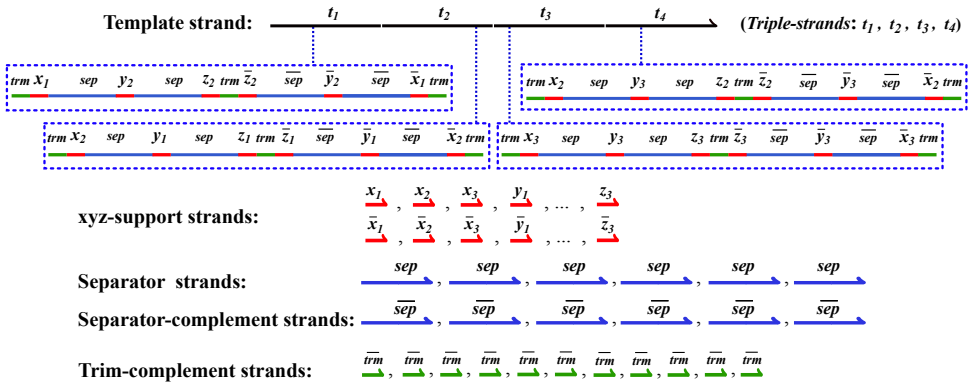
▶ **Definition 8.** *A set $\mathcal{S}$ of strands is* k-robust *if the following properties hold:*

1. *All strands of $\mathcal{S}$ have the same length.*
2. *All strands of $\mathcal{S}$ have at least k A's and at least k T's.*
3. *For any a and b in the set, the unpairedness of optimal structures for a, $\bar{a}$, (a,b), $(\bar{a},\bar{b})$, and $(a,\bar{b})$ is at least k.*

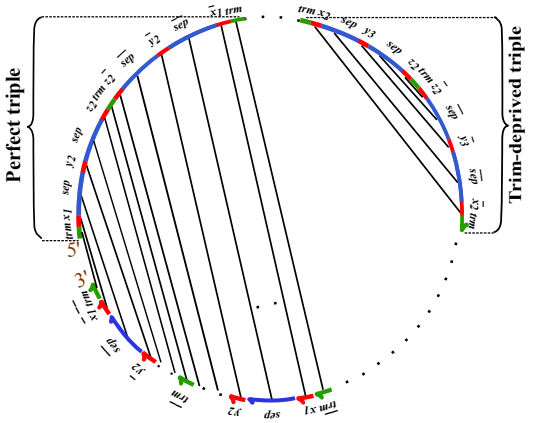
▶ **Theorem 9.** *Let $w > 0$. For any n, a $\log_2 n$-robust set of at least wn strands, each of length $p\log_2 n$ for some constant p, can be designed in $2^{O(\log n)}$ time.*

Proof. Using Theorem 4, for any $w > 0$ and $\delta = 6$ we can obtain, in time $2^{O(\log n)}$, a set $\mathcal{S}'$ of at least wn strands, each of length $k\log_2 n$ for some constant k, such that the insertion-deletion distance between any pair of strands in $\mathcal{S}'$ is at least $6\log_2 n$. Moreover, all strands in $\mathcal{S}'$ have at least $3\log_2 n$ A's and at least $3\log_2 n$ T's. This latter property implies that the strands in $\mathcal{S}'$ have length at least $6\log_2 n$.

Apply the padding function ρ^5 to strands in $\mathcal{S}'$ to obtain a new set $\mathcal{S}$. The strands in $\mathcal{S}$ have length $6k\log_2 n$, which must be at least $36\log_2 n$. Lemma 6 shows that the insertion-deletion distance between any pair of strands in $\mathcal{S}$ is at least $\delta\log_2 n/2 = 3\log_2 n$. Lemma 7 then shows that if a and b are any two strands in the set $\mathcal{S}$, the unpairedness of the optimal structure of a, or its complement, or of (a,b), $(a,\bar{b})$ or $(\bar{a},\bar{b})$, is at least $\min\{\frac{1}{3}|a|, \frac{2}{3}(|a|+|b|), \frac{1}{3}\mathbf{d_{LCS}}(a,b)\}$. Given that $|a|$ and $|b|$ are at least $36\log_2 n$ and that $\mathbf{d_{LCS}}(a,b) = 3\log_2 n$, this lower bound is at least $\log_2 n$. Therefore, the unpairedness of the set $\mathcal{S}$ is at least $\log_2 n$, as desired. ◀



(a)



(b)

Figure 4 Example of the reduction from the 3DM(3) instance of Figure 2. (a) Strands of the resulting MULTI-PKF-SSP instance, specified at the domain level. (b) Partial MFE structure of the strands. Here, the structure involving triple-strand t_1, labeled as *perfect triple*, indicates that the triple (x_1, y_2, z_2) is in the solution of the 3DM(3) instance. Triple-strand t_4 is a *trim-deprived triple* since there are no bonds to bases in the middle trim domain. This structure indicates that triple (x_2, y_3, z_2) isn't selected in the solution.

4 The reduction

We show a polynomial time reduction from 3DM(3) to Multi-Pkf-SSP. Given an instance $I = (X, Y, Z, \mathcal{T})$ of 3DM(3), where $m = |\mathcal{T}|$ and $n = |X| = |Y| = |Z|$, we construct an instance I' of Multi-Pkf-SSP as follows.

Domains used in strands of I'

The strands of I' contain the following domains.

- One domain for each $x \in X$, $y \in Y$, and $z \in Z$ and one domain for each complement. Where no confusion arises, we use x, $\bar{x}$, y, $\bar{y}$, z, and $\bar{z}$ to refer to these domains.
- A *separator* and a *separator-complement* domain, denoted by `Sep` and $\overline{\texttt{Sep}}$.
- A *trim* domain and a *trim-complement* domain, denoted by `Trm` and $\overline{\texttt{Trm}}$ respectively.

Strands of I'

Instance I' consists of the following strands, where each strand is a sequence of domains.

- *Template strand*: One strand that is the concatenation of *triples*. There is one triple for each $(x, y, z) \in \mathcal{T}$, which is the following concatenation of domains:
  $$\texttt{Trm}\ x\ \texttt{Sep}\ y\ \texttt{Sep}\ z\ \texttt{Trm}\ \bar{z}\ \overline{\texttt{Sep}}\ \bar{y}\ \overline{\texttt{Sep}}\ \bar{x}\ \texttt{Trm}$$
  We call the substrands $x\ \texttt{Sep}\ y\ \texttt{Sep}\ z$ and $\bar{z}\ \overline{\texttt{Sep}}\ \bar{y}\ \overline{\texttt{Sep}}\ \bar{x}$ of a triple the $5'$ and $3'$ *flanks*, respectively. We call the `Trm` domains at the ends of the triple the end-trims and the middle `Trm` domain the center-trim.
- *Separator (-complement) support strands*: $2n$ `Sep` strands and another $2n$ $\overline{\texttt{Sep}}$ strands.
- `xyz`*-support strands*: For each x, y and z domain, one strand consisting of just that domain and one for its complement, for a total of $6n$ strands.
- *Trim-complement strands*: $2m + n$ copies of $\overline{\texttt{Trm}}$, which is the complement of the trim domain `Trm`.

We refer to the `xyz`-support strands and the separator and separator-complement support strands collectively as the support strands.

▶ **Lemma 10.** *The total number of support strands is* $10n$.

Proof. This follows immediately from the fact that there are $6n$ `xyz`-supports and $4n$ separator and separator-complement strands in total. ◀

This completes the description of the reduction at the domain level of detail. Figure 4 (a) shows the resulting Multi-Pkf-SSP instance, specified at the domain level, after a reduction from the 3DM(3) instance of Figure 2.

The MFE structure of the resulting set of strands is partially depicted in Figure 4 (b). All domains of the substrand labeled as "perfect triple" are bound to their complements, indicating that the triple (x_1, y_2, z_2) is selected in the solution of the 3DM(3) instance, consistent with the solution shown in Figure 2 (b). The other triple that is depicted is a "trim-deprived triple". This is a triple in which at least one trim domain is unbound. The corresponding triple (x_2, y_3, z_2) does not appear in the solution from Figure 2 (b). Intuitively, there is a trim-complement strand available to bind with each of the $2m$ end-trim domains at the ends of all triples, and in addition the number of xyz-support, separator supports and additional trim-complement strands is necessary and sufficient to have n "perfect triples" in an optimal secondary structure when the 3DM(3) instance has a perfect matching of size n.

Sequence design for I'

To complete the reduction, we specify a sequence design for each domain of I'. For the x, y, and z domains, we use the set of sequences of Theorem 9 with $w = 3$, since we need $3n$ domains (plus their complements) in total. Let E ($= \Theta(\log_2 n)$) be the length of these domains. The trim domain $\texttt{Trm} = \texttt{G}^E$, i.e., consisting of E copies of G, and $\overline{\texttt{Trm}} = C^E$. The Sep domain is $\texttt{A}^{6E}$, and the $\overline{\texttt{Sep}}$ domain is the complement of the Sep domain, namely $\texttt{T}^{6E}$.

The sequence design has the property that there are an equal number of A and T bases overall, since for every x, y, z or separator domain there is a corresponding complementary domain. The total number of C's in trim-complement strands is $(2m + n)E$. The total number of G's in end-trims and center-trims is $3mE$. Since $m \geq n$, the total number of G's is at least as great as the total number of C's. Therefore, under the assumption that only Watson-Crick base pairs can form, the maximum number of base pairs is limited to the total number of A (or T) bases plus the total number of C bases. Let P denote this quantity.

The instance I' is comprised of the strands of I' and the positive integer P.

▶ **Lemma 11.** *Instance I' can be constructed in time polynomial in n.*

5 Reduction correctness

We show that if the given instance I of 3DM(3) has a perfect matching then the optimal secondary structure formed from strands in I' is a single complex that has the maximum possible number P of base pairs. We also show that if the optimal matching of I has size $n - i$ then the optimal structure has only $P - \Omega(iE)$ base pairs.

▶ **Lemma 12.** *If I has a perfect matching, then the strands of I' can form a pseudoknot-free secondary structure, consisting of a single complex and P base pairs, with n perfect triples.*

Proof. Here, in the reduced instance I', bases in the n triples corresponding to the perfect matching can be bound to the corresponding support strands, to form n perfect triples. The end-trims of the remaining triples can also be bound to two trim-complement strands, while their complementary 5′ and 3′ flanks are paired together to make trim-deprived triples. Therefore, as all A's and C's are paired in this single (connected) complex, the number of base pairs is P. ◀

We next consider the case that the optimal matching of I has size at most $n - i$. Let $\text{Opt}(I')$ denote an optimal pseudoknot-free structure of the reduced instance I'. We establish properties that must hold true of $\text{Opt}(I')$ and conclude that when the optimal matching of I has size at most $n - i$, then $\text{Opt}(I')$ has $P - \Omega(iE)$ base pairs.

With respect to a given structure, we say that a domain is *bound* if at least one of its bases forms a base pair. A domain d in a triple (as part of the template strand) is *connected to* a non-template strand s if there is a sequence of non-template strands $s_1, s_2, \ldots, s_j$ where $s_j = s$, such that d forms a base pair with s_1, s_1 forms a base pair with s_2, and so on up to s_{j-1} forming a base pair with $s_j = s$.

We partition the triples into four types, depending on the structure they form in $\text{Opt}(I')$.

- *Perfect triples*: The triple binds to the set of non-template strands that are complementary to the triple domains. (This set of non-template strands contains two Sep's, two $\overline{\texttt{Sep}}$'s, three $\overline{\texttt{Trm}}$'s and six xyz-support strands in total.) The set of perfect triples corresponds to a matching of instance I.
- *Trim-deprived triples*: At least one trim of a triple is unbound.

- *Hogger triples*: These are triples which are not trim-deprived, and moreover, the ten domains in the flanks of a hogger triple are bound to, or connected to, at least eleven support strands in total.
- *Flawed triples*: None of the above. In particular, flawed triples are not trim-deprived.

Since neither hogger nor flawed triples are trim-deprived, the support domains that are bound to or connected to either their $5'$ or $3'$ flanks cannot bind to other domains on the template strand, or a pseudoknot would form.

▶ **Lemma 13.** *The total number of trim-deprived and flawed triples in Opt(I') is at least* $(m-n)+i/11$.

▶ **Lemma 14.** *Either Opt(I') has at least* $m-n+i/22$ *trim-deprived triples, or at least* $i/22$ *flawed triples.*

Proof. By Lemma 13, the total number of trim-deprived and flawed triples is at least $(m-n)+i/11$. So if the number of trim-deprived triples is less than $m-n+i/22$, then the number of flawed triples must be at least $i/22$. ◀

We now adapt our notion of unpairedness from Section 3 to ACT-unpairedness. Let a and b be strands and let $S(a)$ and $S(a,b)$ be secondary structures for strand a and pair (a,b) respectively. The ACT-unpairedness of $S(a)$ or $S(a,b)$ is the number of A, C and T bases that are not paired in $S(a)$ or $S(a,b)$, respectively.

▶ **Lemma 15.** *If the number of trim-deprived triples in Opt(I') is at least* $m-n+i/22$*, then at least* $iE/22$ C*'s are unpaired in Opt(I'), and so Opt(I') has* ACT*-unpairedness* $\Omega(iE)$.

In order to show that many flawed triples cause Opt(I') to have high ACT-unpairedness, we first derive some useful properties about flawed triples. In what follows, we let $F_{5'} = x\ \mathtt{Sep}_{xy}\ y\ \mathtt{Sep}_{yz}\ z$ and $F_{3'} = \bar{z}\ \overline{\mathtt{Sep}}_{yz}\ \bar{y}\ \overline{\mathtt{Sep}}_{xy}\ \bar{x}$ denote the sequences on the $5'$ and $3'$ flanks of a given flawed triple. Let $\mathcal{S}_{5'}$ and $\mathcal{S}_{3'}$ be the sets of support strands that are bound to, or connected to, domains of $F_{5'}$ and $F_{3'}$ respectively, in the structure Opt(I'). The sets $F_{5'}$ and $F_{3'}$ are disjoint, since something is bound to the middle trim in Opt(I'), and the structure has no pseudoknots. Since a flawed triple has at most ten support strands bound to it, either $|\mathcal{S}_{5'}| \leq 5$ or $|\mathcal{S}_{3'}| \leq 5$. In the following lemmas, for concreteness, we suppose that $|\mathcal{S}_{5'}| \leq 5$; the argument when $|\mathcal{S}_{3'}| \leq 5$ is obtained by replacing domains and strands with their complements and bases A and T with each other. Let Opt($F_{5'}$) be the substructure of Opt(I') formed by the bases in $F_{5'}$ and the strands in $\mathcal{S}_{5'}$.

▶ **Lemma 16.** *Suppose that there are* $l \geq 2$ *bonds between one of the x, y or z domains of $F_{5'}$ and either* $\mathtt{Sep}_{xy}$ *or* $\mathtt{Sep}_{yz}$*. Then Opt($F_{5'}$) has* ACT*-unpairedness at least* $5(l-1)$.

▶ **Lemma 17.** *Suppose that in Opt($F_{5'}$),* $|\mathcal{S}_{5'}| \leq 5$ *and the* ACT*-unpairedness of $F_{5'}$ is less than* $(\log_2 n)/3$*. Then the following must hold.*

1. *Each* Sep *domain of $F_{5'}$ is bound to a* $\overline{\mathtt{Sep}}$*-support domain.*
2. *Each x, y and z domain of $F_{5'}$ is bound to an* xyz*-support domain.*

As a consequence, each x, y, and z domain of $F_{5'}$ is bound to a distinct xyz*-support of $\mathcal{S}_{5'}$, each* Sep *domain of $F_{5'}$ is bound to a distinct* $\overline{\mathtt{Sep}}$ *support of $\mathcal{S}_{5'}$, and $\mathcal{S}_{5'}$ contains exactly three* xyz*-supports and two* $\overline{\mathtt{Sep}}$ *supports.*

▶ **Lemma 18.** *Let* $F_{5'} = x\ \mathtt{Sep}_{xy}\ y\ \mathtt{Sep}_{yz}\ z$ *be the left flank of a flawed triple. Suppose that in Opt($F_{5'}$),* $|\mathcal{S}_{5'}| \leq 5$*. Then for any constant* $\alpha < 1/7$*, the* ACT*-unpairedness of Opt($F_{5'}$) is at least* $\alpha \log_2 n$.

▶ **Lemma 19.** *If the optimal matching of I has size at most $n-i$, then Opt(I') has $P-\Omega(iE)$ base pairs.*

Proof. By Lemma 14, Opt(I') either has at least $m-n+i/22$ trim-deprived triples, or at least $i/22$ flawed triples.

First suppose that Opt(I') has at least $m-n+i/22$ trim-deprived triples. Then by Lemma 15, Opt(I') has ACT-unpairedness $\Omega(iE)$. Similarly, if Opt(I') has at least $i/22$ flawed triples, then by Lemma 18, each flawed triple has ACT-unpairedness $\Omega(\log n)=\Omega(E)$, since $E=\Theta(\log n)$. Again, the total ACT-unpairedness is $\Omega(iE)$.

Recall that all A's, C's and T's must be paired in order for the total number of base pairs to be P. Since the total ACT-unpairedness is $\Omega(iE)$, it must be that the number of base pairs in Opt(I') is at most $P-\Omega(iE)$. ◀

▶ **Theorem 20.** *Multi-Pkf-SSP is* NP*-complete.*

Proof. Let I be any instance of Multi-Pkf-SSP, i.e, m nucleic acid strands and a positive integer k. Given a certificate for I, which includes a secondary structure S and an ordering of the m strands, we can check in time polynomial in the total length of the strands whether S is a valid, pseudoknot-free secondary structure and whether it has k base pairs. Therefore, Multi-Pkf-SSP is in NP.

Moreover, in the last section we provided a polynomial time reduction from any instance I of 3dm(3) to an instance I' of Multi-Pkf-SSP. The optimal structure Opt(I') has P base pairs if I has a perfect matching, by Lemma 12, and Opt(I') has less than P base pairs if I does not have a perfect matching (by Lemma 19), where P is the total number of A, T and C bases of the strands of instance I'.

Putting these together, we conclude that Multi-Pkf-SSP is NP-complete. ◀

Until now, we have only considered the number of base pairs in the MFE structure under the assumption that there is no penalty for strand association, i.e., $K_{\text{assoc}}=0$. Our construction has the property that structure Opt(I') is a single complex when I has a perfect matching. When $K_{\text{assoc}}>0$ the penalty to bring the $2m+11n+1$ strands into a single complex is $(2m+11n)K_{\text{assoc}}$. However, the number of base pairs formed between complementary domains of distinct strands is at least E, where $E=\Theta(\log n)$. Thus, for any positive constant K_{assoc} the value of E can be scaled by a constant to ensure that a single domain binding is always favourable, even when decreasing the number of complexes by one.

6 Approximability

We proved that the Multi-Pkf-SSP problem is NP-complete in Theorem 20. Given this result, it is natural to investigate whether there is a polynomial time algorithm to approximate the optimal secondary structure of multi-stranded systems. We show in Theorem 22 that the Max-Multi-Pkf-SSP problem is in fact APX-hard. This result asserts that there is no polynomial time approximation scheme (PTAS) for this problem, unless P = NP.

We first note in Lemma 21 that our reduction of Section 4 is approximation-preserving, transforming one optimization problem into another one. We then prove that this construction also maps a solution of Max-Multi-Pkf-SSP to a solution of Max-3dm(3).

▶ **Lemma 21.** *Our reduction from an instance I of Max-3dm(3) to an instance I' of Max-Multi-Pkf-SSP has the following properties:*

- *If I has a matching of size n then $|Opt(I')|=P$.*
- *If I has a matching of size at most $(1-\epsilon_0)n$ then $|Opt(I')|\leq P-\alpha\epsilon_0 nE$, for some constant $\alpha>0$.*

Proof. This lemma directly follows from Lemmas 12 and 19. ◀

▶ **Theorem 22.** *Max-Multi-Pkf-SSP is* APX*-hard.*

Proof. Let I' be an instance of Max-Multi-Pkf-SSP obtained from an instance I of Max-3dm(3) where the size of the three sets is n and there are m total triples. First, we review the quantities E and P of the reduction of Section 4. Recall that $E = \Theta(\log n)$ specifies the lengths of xyz-support domains in instance I' obtained from instance I. Our sequence design and Lemma 11 ensure that instance I' has $\Theta(n) + \Theta(m)$ domains of length $\Theta(E)$. Recall that P is the total number of base pairs in an optimal structure for I' if I has a perfect matching. A perfect matching for I would have n triples. It follows that $P = \Theta(n \log_2 n)$.

We now apply Lemma 21 to show APX-hardness of Max-Multi-Pkf-SSP. Suppose to the contrary that for some $\epsilon > 0$, there is a $(1-\epsilon)$–approximation algorithm for this problem. Then, the following hold:

- If I of Max-3dm(3) has a matching of size n, then on instance I' the algorithm returns a solution with value at least $(1-\epsilon)|\text{Opt}(I')| = (1-\epsilon)P$.
- If instance I has a matching of size at most $(1-\epsilon_0)n$, then on instance I' the algorithm returns a solution with value at most $|\text{Opt}(I')| \leq P - \alpha\epsilon_0 nE$.

Therefore, if $P - \alpha\epsilon_0 nE < (1-\epsilon)P$ the algorithm can distinguish between the cases where I has a matching of size n or of size at most $(1-\epsilon_0)n$. By our current assumptions about P and E, the above inequality holds if $\epsilon < \alpha\epsilon_0 nE/P$. This contradicts Theorem 2, on the APX-hardness of Max-3dm(3). ◀

7 Conclusions

This work resolves an open question on algorithms for pseudoknot-free secondary structure prediction of nucleic acids: Can we efficiently compute the minimum free energy (MFE) pseudoknot-free secondary structure for a multi-set of DNA or RNA strands? We have shown that this problem is NP-hard, and is therefore computationally intractable, unless P = NP. A natural question then is whether solutions to the problem can be efficiently approximated, if P ≠ NP. Unfortunately, there is a limit to the accuracy of any such method. We have shown that the optimization problem of finding the MFE structure for a multi-set of nucleic acid strands is hard for the complexity class APX, the class of NP optimization problems that have constant factor approximation algorithms. The result implies that there does not exist a polynomial time approximation scheme for this problem, unless P = NP. Given these results, it suggests that heuristic methods, such as stochastic local search, and randomized algorithms should be investigated for structure prediction of multiple interacting strands.

References

1 Tatsuya Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots,. *Discrete Applied Mathematics*, 104(1-3):45–62, August 2000.

2 Mirela Andronescu, Zhi Chuan Zhang, and Anne Condon. Secondary structure prediction of interacting RNA molecules. *Journal of Molecular Biology*, 345(5):987–1001, February 2005.

3 Karl Bringmann, Fabrizio Grandoni, Barna Saha, and Virginia Vassilevska Williams. Truly Subcubic Algorithms for Language Edit Distance and RNA-Folding via Fast Bounded-Difference Min-Plus Product. *SIAM Journal on Computing*, 48(2):481–512, 2019.

4 Robert M. Dirks, Justin S. Bois, Joseph M. Schaeffer, Erik Winfree, and Niles A. Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Rev.*, 49(1):65–88, 2007. doi:10.1137/060651100.

5 Mark E Fornace, Nicholas J Porubsky, and Niles A Pierce. A unified dynamic programming framework for the analysis of interacting nucleic acid strands: Enhanced models, scalability, and speed. *ACS Synthetic Biology*, 9(10):2665–2678, 2020.

6 Kenichi Fujibayashi, Rizal Hariadi, Sung Ha Park, Erik Winfree, and Satoshi Murata. Toward reliable algorithmic self-assembly of DNA tiles: A fixed-width cellular automaton pattern. *Nano Letters*, 8(7):1791–1797, July 2008.

7 Michael R Garey and David S Johnson. Computers and Intractability: A guide to NP-completeness, 1979.

8 Dan S Hirschberg. *Pattern matching algorithms*, chapter Serial computations of Levenshtein distances, pages 123–142. Oxford university press, 1997.

9 J. A. Jaeger, D. H. Turner, and M. Zuker. Predicting optimal and suboptimal secondary structure for RNA. *Methods in Enzymology*, 183:281–306, 1990.

10 J. Justesen. A class of constructive asymptotically good algebraic codes. *Information Theory, IEEE Transactions on*, 18(5):652–656, 1972.

11 Viggo Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37(1):27–35, 1991.

12 Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology : AMB*, 6:26, November 2011.

13 R. B. Lyngsøand C. N. Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of Computational Biology*, 7(3-4):409–427, 2000.

14 Rune B. Lyngsø. Complexity of pseudoknot prediction in simple models. In Josep Diaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannellã, editors, *Proceedings, Automata, Languages and Programming 31st Internationa l Colloquium, ICALP*, volume 3142 of *Lecture Notes in Computer Science*, pages 919–931. Springer Berlin/Heidelberg, January 2004. doi:10.1007/b99859.

15 David H. Mathews and Douglas H. Turner. Prediction of RNA secondary structure by free energy minimization. *Current Opinion in Structural Biology*, 16(3):270–278, 2006. doi:10.1016/j.sbi.2006.05.010.

16 J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, June 1990.

17 R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 77(11):6309–6313, November 1980.

18 R. Nussinov, G. Pieczenik, J. Griggs, and D. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35(1):68–82, 1978.

19 L.J. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45(7):2552–2557, 1999. doi:10.1109/18.796406.

20 Bryan Wei, Mingjie Dai, and Peng Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485(7400):623–626, 2012.

21 S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, February 1999.

22 Joseph N. Zadeh, Brian R. Wolfe, and Niles A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452, 2011. doi:10.1002/jcc.21633.

23 M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, January 1981.

24 Michael Zuker and David Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, July 1984. doi:10.1007/BF02459506.

A Technical Appendix

A.1 Proof of Lemma 6

Proof. Let $\mathbf{d_{LCS}}(a,b) = k$. We suppose that $\mathbf{d_{LCS}}(\rho^i(a), \rho^i(b)) < k/2$ and obtain a contradiction. Let $\mathcal{A}$ be an optimal alignment of $\rho^i(a)$ and $\rho^i(b)$. Throughout, when referring to characters in $\rho^i(a)$ and $\rho^i(b)$, we denote the characters of the original strings a and b by $\mathtt{A}_o$ and $\mathtt{T}_o$ and the padded $\mathtt{A}$'s by $\mathtt{A}_p$. Each pair of characters in alignment $\mathcal{A}$ has one of four types: *original*, with two original characters; *padded*, with two padded characters; *mixed*, with one A_o and one A_p, or *gap*, with one gap symbol. Let n be the length of a and b, and let $\#_{orig}$, $\#_{pad}$ and $\#_{mix}$ denote, in order, the counts of original, padded and mixed pairs, respectively. To prove the lemma, we establish various bounds on these counts, as a function of n and k.

First, note that any alignment of $\rho^i(a)$ and $\rho^i(b)$ has at most $n - \frac{k}{2}$ original pairs: Otherwise, we could use the alignment to obtain an alignment of a and b with less than k gap pairs, which is not possible since $\mathbf{d_{LCS}}(a,b) = k$. Therefore,

$$\#_{orig} \leq n - \tfrac{k}{2}. \tag{1}$$

Second, using Theorem 3 and our assumption that $\mathbf{d_{LCS}}(\rho^i(a), \rho^i(b)) < \frac{k}{2}$, we have that $\mathbf{LCS}(\rho^i(a), \rho^i(b)) \geq (i+1)n - \lfloor \frac{k}{4} \rfloor$, and so

$$\#_{orig} + \#_{pad} + \#_{mix} = \mathbf{LCS}(\rho^i(a), \rho^i(b)) \geq (i+1)n - \lfloor \tfrac{k}{4} \rfloor. \tag{2}$$

Third, we'll obtain a lower bound on $\#_{orig}$. Note that $2\#_{pad} + \#_{mix}$ is upper bounded by the total number of $\mathtt{A}_p$ characters, and so is at most $2in$. Therefore $\#_{pad} + \lceil \frac{\#_{mix}}{2} \rceil \leq in$. Substituting this inequality into Equation 2, we have that

$$\#_{orig} \geq n - \lfloor \tfrac{k}{4} \rfloor - \lfloor \tfrac{\#_{mix}}{2} \rfloor. \tag{3}$$

Finally, from inequalities 1 and 3 we have $\#_{mix} \geq k/2$.

We now partition the mixed pairs into two types: sloppy and tight.

- A mixed pair p of alignment $\mathcal{A}$ is *sloppy* if, among the first i pairs to the right of p, there is at least one gap pair containing a T_o or A_p character. Mixed pairs must be separated by at least i pairs since there are at least i A_p's between any two A_o', and so the gap pairs corresponding to each of the sloppy mixed pairs are distinct. From property (3) above, $\mathbf{d_{LCS}}(\rho^i(a), \rho^i(b))$ is equal to the number of gap pairs in $\mathcal{A}$. Since we are assuming that $\mathbf{d_{LCS}}(\rho^i(a), \rho^i(b)) < \frac{k}{2}$, the alignment $\mathcal{A}$ has less than $\frac{k}{2}$ gapped pairs, and thus has less than $\frac{k}{2}$ sloppy mixed pairs.
- If p is not sloppy, we call it *tight*. Since less than $\frac{k}{2}$ of the mixed pairs are sloppy, at least $\#_{mix} - \frac{k}{2} + 1$ of the mixed pairs are tight.

If p is tight, let p' be the first pair to the right of p that is not a padded pair. Such a pair p' must exist, since our padding function is such that any A_p character is eventually followed by an original character. Pair p' is either a gap pair containing A_o or is a mixed pair, in which case it also contains A_o. In either case, because exactly i A_p's separate any two original characters, if the A_o character of pair p is in string a then the A_o character of pair p' is in string b and vice versa. In what follows, we refer to p' as p's *partner*. Note that p' may itself be a tight pair.

Using these $\#_{mix} - \frac{k}{2} + 1$ tight pairs, we now convert alignment $\mathcal{A}$ to another alignment $\mathcal{A}'$ with at least $n - \frac{k}{2} + 1$ original pairs, obtained as follows; see example in Figure 5. Starting from the leftmost pair of alignment $\mathcal{A}$ and working towards the right, find the first tight

	p			p'						
$\ldots$	$\mathbf{A_o}$	A_p	A_p	A_p	A_p	A_p	T_o	A_p	A_p	$\ldots$
$\ldots$	A_p	A_p	A_p	$\mathbf{A}_o$	A_p	A_p		A_p	A_p	$\ldots$

(a) Alignment $\mathcal{A}$.

$\ldots$				$\mathbf{A_o}$	A_p	A_p	A_p	A_p	A_p	T_o	A_p	A_p	$\ldots$
$\ldots$	A_p	A_p	A_p	$\mathbf{A_o}$				A_p	A_p		A_p	A_p	$\ldots$

(b) Alignment $\mathcal{A}'$.

Figure 5 Illustration of the construction of Lemma 6. In alignment $\mathcal{A}$, the pair at position p is a tight pair; its partner is at position p' and is a sloppy pair. Alignment $\mathcal{A}'$ has one more original pair, indicated in bold, than does alignment $\mathcal{A}$.

mixed pair p of $\mathcal{A}$ and its partner p'. Remove p, p' and all of the intervening (padded) pairs between them from the alignment, and instead pair each padded character from the removed pairs with a gap, and pair the A_o character of p with the A_o character of p' (recall that one of these A_o characters is in string a and the other is in string b). Repeat, starting from the pair just to the right of p', until the rightmost end of $\mathcal{A}$ is reached.

The number of new original pairs obtained in this manner is at least $\lfloor \frac{\#_{mix}}{2} \rfloor - \lceil \frac{k}{4} \rceil + 1$. To see why:

- If $\#_{mix} - \frac{k}{2} + 1$ is odd, then the number of new original pairs is at least

$$(\#_{mix} - \tfrac{k}{2})/2 + 1 \geq \lfloor \tfrac{\#_{mix}}{2} \rfloor - \lceil \tfrac{k}{4} \rceil + 1.$$

  This lower bound is achieved when all but one of the tight mixed pairs are partners.
- If $\#_{mix} - \frac{k}{2} + 1$ is even, then the number of new original pairs is at least

$$(\#_{mix} - \tfrac{k}{2} + 1)/2 = \lfloor (\#_{mix} - \tfrac{k}{2})/2 \rfloor + 1/2 + 1/2 = \lfloor \tfrac{\#_{mix}}{2} \rfloor - \lfloor \tfrac{k}{2}/2 \rfloor + 1.$$

  This lower bound is achieved when all partners are themselves tight mixed pairs.

Therefore, the number of original pairs in alignment $\mathcal{A}'$ is

$$\begin{aligned} \#_{orig} + \quad & \lfloor \tfrac{\#_{mix}}{2} \rfloor - \lceil \tfrac{k}{4} \rceil + 1 \\ & \geq n - \lfloor \tfrac{k}{4} \rfloor - \lfloor \tfrac{\#_{mix}}{2} \rfloor + \lfloor \tfrac{\#_{mix}}{2} \rfloor - \lceil \tfrac{k}{4} \rceil + 1 \quad \text{(using inequality 3)} \\ & = n - \tfrac{k}{2} + 1. \end{aligned}$$

But as noted earlier, any alignment of $\rho^i(a)$ and $\rho^i(b)$ has at most $n - \frac{k}{2}$ original pairs, and so we have our contradiction. ◀

A.2 Proof of Lemma 7

Proof. To show part 1, first suppose that s is a substrand of $a = \rho^5(a')$. If $|s| \leq 2$, then no bases of s are paired in $S(s)$, given our assumption that consecutive bases in a strand cannot form a base pair, and so part 1 holds. If $|s| \geq 3$, the number of (intra-molecular) base pairs of $S(s)$ is at most the number of `T`'s in s. If $3 \leq |s| \leq 6$ then s can have at most one `T`, and thus at most one base pair, so s has at least $|s| - 2$ unpaired bases and again part 1 holds. Suppose that $|s| \geq 7$. Because s is a substrand of a padded strand, the number of `T`'s in s is at most $\lceil 2|s|/7 \rceil$: this maximum is achieved if $|s| = 7$ and s both starts and ends with a `T`. Even if all of the `T`'s of s are paired to `A`'s, the number of unpaired `A`'s is still at least $\lfloor 3|s|/7 \rfloor \geq |s|/3$ since $|s| \geq 7$. The argument when s is a substrand of $\bar{a}$ is obtained by replacing `A`'s with `T`'s in the argument for a substrand of a.

Similarly, the total number of T's in $S(a,b)$ is at most $(|a|+|b|)/6$ and so the unpairedness is at least $4(|a|+|b|)/6$. The argument for the unpairedness of $S(\bar{a},\bar{b})$ is obtained by replacing A's with T's in the argument for $\{a,b\}$.

Finally, the inter-molecular base pairs of $S(a,\bar{b})$ correspond to a common subsequence of strands a and b, and thus the number of such base pairs is at most $\mathbf{LCS}(a,b) = n - \frac{\mathbf{d_{LCS}}(a,b)}{2}$ by Theorem 3. Therefore the total number of bases in both a and $\bar{b}$ that do not form inter-molecular base pairs of $S(a,\bar{b})$ is at least $\mathbf{d_{LCS}}(a,b)$. Now consider any substructure of $S(a,\bar{b})$ within some maximal substrand s of either a or $\bar{b}$ that has no inter-molecular base pairs. The unpairedness of this substructure is at least $\frac{1}{3}|s|$, by part 1 of this Lemma. Thus, over all substrands that do not contain inter-molecular base pairs, at least a fraction $\frac{1}{3}$ of bases are unpaired (not involved in intra-molecular base pairs). Since the total length of such substrands is at least $\mathbf{d_{LCS}}(a,b)$, the unpairedness of $S(a,\bar{b})$ is at least $\frac{1}{3}\mathbf{d_{LCS}}(a,b)$. ◀

A.3 Proof of Lemma 11

Proof. Instance I' has one template strand, $2n$ separator supports, $2n$ separator-complement supports $6n$ xyz-support strands, and $2m+n$ trim-complement strands, for a total of $2m+11n+1$ strands. The template strand has $13m$ domains and the other strands have one domain each, for a total of $15m+11n$ domains.

Since every domain in the construction has length $\Theta(\log_2 n)$, instance I' is of size polynomial in n overall. The sequences can also be designed in polynomial time: The sequence design of separator and trim domains is trivial, and the sequences for the x, y, z domains can be designed in time polynomial in n by Theorem 9. ◀

A.4 Proof of Lemma 13

Proof. The number of trim-deprived and flawed triples is $m-p-h$, where m, p, and h are the number of triples, perfect triples, and hogger triples, respectively.

Perfect triples and hogger triples are not trim-deprived. Therefore, any support strand connected to a perfect triple or a hogger triple cannot also be connected to another triple without creating a pseudoknot. Each perfect triple has 10 support strands bound to it, and each hogger triple has at least 11 connected support strands. From Lemma 10, there are $10n$ support strands in total, so $10p+11h \leq 10n$ and

$$h \leq 10(n-p)/11.$$

Since the optimal matching of I has size at most $n-i$, the number of perfect triples p must be at most $n-i$ and so $n-p \geq i$. Therefore, the total number of trim-deprived and flawed triples is

$$m-p-h \geq m-p-10(n-p)/11 = m-n+(n-p)/11 \geq (m-n)+i/11.$$ ◀

A.5 Proof of Lemma 15

Proof. Each trim-deprived triple forms at most $2E$ CG base pairs, with the G's being in the trims (center-trim and end-trims) of the triple and the C's being in trim-complement strands. Triples that are not trim-deprived form at most $3E$ CG base pairs. There are no other CG base pairs. So, the total number of CG base pairs is at most

$$(m-n+i/22)2E + (m-(m-n+i/22))3E = (2m+n-i/22)E.$$

The total number of trim-complement strands is $2m+n$, each containing E C's. So, the number of unpaired C bases in trim-complements is at least $iE/22$. ◀

A.6 Proof of Lemma 16

Proof. Suppose that there are l bonds between x and $\texttt{Sep}_{xy}$; the other cases are similar. Since $\texttt{Sep}_{xy}$ contains only A's, only T's of x can bind with $\texttt{Sep}_{xy}$. Our sequence design ensures that there are at least five padded A's between any two successive T's of x. Therefore, in order to avoid pseduoknots, if there are l bonds between x and $\texttt{Sep}_{xy}$, at least $5(l-1)$ padded A's remain unpaired. ◀

A.7 Proof of Lemma 17

Proof. Suppose to the contrary that the first condition does not hold, i.e., one of $F_{5'}$'s Sep domains is not bound to a $\overline{\texttt{Sep}}$ support. The total number of T's that can bind to the Sep domain is at most $5.5E$, accounted for as follows. There are at most $3E/6$ T's in the x, y, and z domains of $F_{5'}$ plus at most $5E$ in the remaining support strands, if there are five xyz-support strands. Thus at least $E/2$ of the $6E$ A's in the Sep domain are unpaired. Since $E \geq \log_2 n$, we get a contradiction to the hypothesis of the lemma. Thus the first condition must hold.

Next suppose that the first condition holds but that the second does not; specifically that the x domain of $F_{5'}$ is not bound to an xyz-support domain (the argument is similar for the y or z domains). Recall that domain x contains at least $\log_2 n$ T's, since by design the domains comprise a $\log_2 n$-robust set. At least $2(\log_2 n)/3$ of the T's must be paired, or the hypothesis of the lemma that the ACT-unpairedness of $F_{5'}$ is less than $(\log_2 n)/3$ would not be true. Since the first condition of the lemma holds, the Sep domain adjacent to x on the $5'$ flank is bound to a $\overline{\texttt{Sep}}$ strand. Therefore domain x cannot have bonds to domain y or z, or to the Sep domain between y and z, or a pseudoknot would form. Also, the T's in domain x cannot bind to $\overline{\texttt{Sep}}$ strands, since $\overline{\texttt{Sep}}$'s are composed only of T's. If there were at least $(\log_2 n)/3$ bonds between x and $\texttt{Sep}_{xy}$, Lemma 16 would imply that x has ACT-unpairedness at least $5((\log_2 n)/3 - 1) \geq \log_2 n$, again contradicting the hypothesis of the lemma.

Therefore, at least $(\log_2 n)/3$ T's of x must form intramolecular bonds with A's that are also in the x domain. The total length of substrands of x that have either unpaired bases or intramolecular base pairs must be at least $3(\log_2 n)/3$: this lower bound is met if each T, say at position i of x is bound to an A that is either at position $i-2$ or $i+2$ (since we assume that no base pair can form between consecutive bases). Part 1 of Lemma 7 therefore implies that x has ACT-unpairedness at least $(\log_2 n)/3$, once again contradicting the hypothesis of the lemma. We conclude that the second condition of the lemma must hold.

Since both conditions hold, it cannot be that two of the x, y, and z domains of $F_{5'}$ are bound to the same xyz-support of $\mathcal{S}_{5'}$, or a pseudoknot would form with bonds between a Sep of $F_{5'}$ and a $\overline{\texttt{Sep}}$ support. Similarly, it cannot be that both Sep's have bonds to the same $\overline{\texttt{Sep}}$. Hence, each Sep domain of $F_{5'}$ is bound to a distinct $\overline{\texttt{Sep}}$ support of $\mathcal{S}_{5'}$, and $\mathcal{S}_{5'}$ contains exactly three xyz-supports and two $\overline{\texttt{Sep}}$ supports, completing the proof of the Lemma. ◀

A.8 Proof of Lemma 18

Proof. Let $\alpha < 1/7$. Suppose to the contrary that the ACT-unpairedness of $\mathrm{Opt}(F_{5'})$ is less than $\alpha \log_2 n$. By Lemma 17, $\mathcal{S}_{5'}$ must contain three xyz-supports, say a, b, and c, with a bound to x, b bound to y, and c bound to x.

We first show that in $\mathrm{Opt}(F_{5'})$, there can be at most $\alpha \log_2 n/5$ bases between a Sep domain of $F_{5'}$ and one of the domains x, y, or z adjacent to the Sep domain. Otherwise, by Lemma 16, at least $\alpha \log_2 n$ bases of a would be unpaired, and we get a contradiction. Similarly, there can be at most $\alpha \log_2 n/5$ bases between a $\overline{\texttt{Sep}}$ domain of $F_{5'}$ and one of the domains a, b, or c adjacent to the $\overline{\texttt{Sep}}$ domain.

Since $F_{5'}$ is the flank of a flawed triple, either $a \neq \bar{x}$, $b \neq \bar{y}$, or $c \neq \bar{z}$. First suppose that $a \neq \bar{x}$. Since the set of domains is $\log_2 n$-robust, there can be at most $E - \log_2 n$ base pairs between a and x. By the argument in the previous paragraph, x has at most $\alpha(\log_2 n)/5$ bases to $\texttt{Sep}_{xy}$. Similarly, if $\overline{\texttt{Sep}}_{ab}$ is the separator complement between a and b, then a has at most $\alpha(\log_2 n)/5$ bases to $\overline{\texttt{Sep}}_{ab}$. If a has base pairs with $\texttt{Sep}_{xy}$, then x cannot have base pairs with $\overline{\texttt{Sep}}_{ab}$ and vice versa, in order to avoid pseudoknots. Therefore, either a or x has at least $\log_2 n - \alpha(\log_2 n)/5 \geq 34(\log_2 n)/35$ bases that are either unpaired or form intramolecular bonds. By Lemma 7, either a or x has unpairedness at least $11(\log_2 n)/35 \geq (\log_2 n)/4$, proving the lemma. The argument when $c \neq \bar{z}$ is similar to that when $a \neq \bar{x}$.

Finally, suppose that $a = \bar{x}$ and $c = \bar{z}$ but $b \neq \bar{y}$. As noted earlier, b has at most $\alpha(\log_2 n)/5$ bonds with each $\overline{\texttt{Sep}}$ adjacent to it. Also, at least $\log_2 n$ bases of b are not paired with y, since the set of domains is $\log_2 n$-robust. Of these, at most $\alpha \log_2 n$ can be unpaired, or again we get a contradiction. Therefore, b has at least $\log_2 n - 2\alpha(\log_2 n)/5 - \alpha \log_2 n = \log_2 n - 7\alpha(\log_2 n)/5$ bonds to the $\texttt{Sep}$'s adjacent to y, and so b has at least $\frac{1}{2}(\log_2 n - 7\alpha(\log_2 n)/5)$ bonds to $\texttt{Sep}_{xy}$.

Moreover, $\overline{\texttt{Sep}}_{ab}$ must have at least $6E - 12\alpha(\log_2 n)/5$ base pairs with $\texttt{Sep}_{xy}$. This is because $\overline{\texttt{Sep}}_{ab}$ has at most $\alpha(\log_2 n)/5$ bases with each of a and b, and $\overline{\texttt{Sep}}_{ab}$ has at most $3\alpha \log_2 n$ bases paired with x. To see why the latter assertion holds, note that otherwise at least $3\alpha \log_2 n$ bases of a are not paired with any strand other than a and thus by Lemma 7, at least $\alpha \log_2 n$ bases of a are unpaired, which again is a contradiction. Therefore, $\overline{\texttt{Sep}}_{ab}$ has at most $(2/5 + 3)\alpha \log_2 n$ pairs in total with a, x, and b, and since at most $\alpha \log_2 n$ bases of $\overline{\texttt{Sep}}_{ab}$ can be unpaired, $\overline{\texttt{Sep}}_{ab}$ has at least $6E - (2/5 + 3 - 1)\alpha \log_2 n = 6E - (12/5)\alpha \log_2 n$ base pairs with $\texttt{Sep}_{xy}$.

Therefore the total number of bases that are paired with bases of $\texttt{Sep}_{xy}$ is at least $\frac{1}{2}(\log_2 n - 7\alpha(\log_2 n)/5)$ (with b) plus $6E - (12/5)\alpha \log_2 n$ (with $\overline{\texttt{Sep}}_{ab}$). The total is

$$6E + (1/2 - 7\alpha/10 - \alpha(12/5)) \log_2 n \geq 6E + (1/2 - \alpha(31/10)) \log_2 n.$$

Since $\alpha \leq 1/7$, this quantity is greater than $6E$, again a contradiction since the length of $\texttt{Sep}_{xy}$ is $6E$. ◀