On the Design of Oligos for Gene Synthesis

Chris Thachuk
School of Computing Science
Simon Fraser University
Burnaby BC, Canada
cthachuk@cs.sfu.ca

Anne Condon
Department of Computer Science
University of British Columbia
Vancouver BC, Canada
condon@cs.ubc.ca

Abstract— **Methods for reliable synthesis of long genes offer great promise for protein synthesis via expression of synthetic genes, with applications to improved analysis of protein structure and function, as well as engineering of novel proteins. Current technologies for gene synthesis use computational methods for design of short oligos, which can then be reliably synthesized and assembled into the desired target gene. For collision-oblivious oligo design – when mishybridizations between oligos are ignored – we give a simple and efficient dynamic programming algorithm. We conjecture that the collision-aware oligo design problem is $\mathcal{NP}$-hard and provide evidence that mishybridizations between oligos occur infrequently in the designs from the collision-oblivious algorithm. We extend our dynamic programing algorithm to achieve collision-aware oligo design, when the target gene can be partitioned into independently-assembled short segments. We evaluate our methods on a large biological gene set.**

I. Introduction

Gene synthesis – efficient construction of long protein-coding or RNA-coding DNA strands – is emerging as an important technology for genomics and synthetic biology. Synthetic genes can be used to express target proteins of interest in a host cell, making it possible to produce protein fragments of a size manageable for structural analysis, to understand how variations in protein sequence affects binding properties of the protein, and to design novel proteins [1]. Use of a designed synthetic gene which codes for the target protein, rather than a naturally occurring gene, can also enhance the gene's expression level in the host, for example by matching the codon bias with that of the host in which the gene is expressed or by removing introns from the gene [2].

Several methods for synthesis of small genes or gene fragments (ranging from less than a hundred up to a thousand or so bases in length) are currently in use. For example, in assembly PCR [3], short oligos are selected which cover the desired gene duplex, with overlaps between successive oligos on the complementary (so-called sense and anti-sense) strands of the duplex. The oligos are synthesized separately, and are pooled in solution. Assembly of the oligos is achieved via hybridization of overlapping oligos on the sense and anti-sense strands. PCR extension is used to fill in any gaps in the assembly, and to amplify the product. Emerging technologies aim to significantly improve the scale and reliability of gene synthesis, enabling synthesis of genes of length 10kb (kilobases) and ultimately up to 100kb, as well as multiplexed synthesis of sets of genes [4]. These technologies typically avoid the traditional high cost of individual oligo synthesis, by the use of parallel synthesis of primer-tagged oligos on photolithographic microarrays, followed by amplification of oligos and cleavage of primers. Further array-based hybridization techniques are used to identify and remove oligos with incorrect base composition due to synthesis errors.

The success of all of these methods relies in part on properties of the selected short oligos. This leads to a computational *oligo design problem*: given an input DNA duplex, select oligos from both the sense and antisense strands, so as to satisfy the following conditions. First, the oligos should *cover* the duplex – if oligos are ordered by distance from one end of the duplex, they should alternate between sense and antisense strands, with some overlap between successive oligos. Second, the oligos should yield *no synthesis and assembly errors*: an oligo should not fold on itself or stably hybridize to any oligo, other than those which it overlaps in the covering of the duplex. As the scale of gene synthesis grows, the computational design of short oligos becomes more critical to the success of gene synthesis technologies. Typically, there is some flexibility in the oligo lengths, and so there are exponentially many possible designs.

There are already many algorithms and software packages available for oligo design [1], [2], [4]–[10]. These methods vary in the types of design criteria they support, and in the underlying design optimization techniques (as well as in other aspects not considered further here, such as the user interface). However, the literature provides little or no insight on the computational complexity of the many interesting variations of the problem of designing oligos for gene synthesis. Moreover, the techniques reported in the literature are all heuristic in nature, but there are no empirically-obtained insights on the relative efficiency of the design techniques used. Because of this, we focus next on the most common design criteria that have been proposed in the literature, other than optimizing codon bias, which we do not consider further in this paper.

To ensure reliable synthesis and assembly, designed oligos which cover a DNA duplex should satisfy one or more of the following criteria:

Limited Length Range [1], [4]–[8], [10]: The length of each oligo should fall within a short specified range (typically between ±4 and ±10 nucleotides from a given length, which can be from 40 to 70 nucleotides); or a maximum length is specified.

Structure Free [2], [4], [5]: Each selected oligo should not self-hybridize to form secondary structure that would interfere with the oligo synthesis.

Uniform melting temperature (Tm) [2], [5], [7]: The Tm of duplexes formed from successive overlapping oligo fragments should fall within a narrow range.

Collision Free [4], [7]: Oligos on one strand of the duplex should bind specifically to overlapping oligos on the complementary strand, and avoid *collisions*, whereby a pair of oligos from different parts of the duplex hybridize stably.

Synthon-informed [2], [5]: Jayaraj et al. [5] propose that assembly of a long gene be done in two phases. First, short oligos are assembled into segments, called synthons. Then, larger sequences (5000 bases) are assembled from synthons. With this approach, selected oligos for the whole gene need not be collision free; rather it is is sufficient that with each synthon, selected oligos are collision free.

We first study the problem of oligo design to satisfy the limited length range, structure free, and uniform Tm criteria. We call this the *collision oblivious* oligo design problem. We show that this problem can be solved efficiently via a simple dynamic programming algorithm. We also consider the *collision aware* oligo design problem, when the collision free criterion is added. We conjecture this problem is $\mathcal{NP}$-hard and previously provided some evidence to this effect by showing that an abstraction of this problem is $\mathcal{NP}$-complete [11]. We extend our dynamic programming algorithm to partition an oligo design into collision free synthons. Finally, we provide an empirical analysis of the degree to which collisions arise in real biological genes, and show that our synthon-informed approach can successfully handle collisions that arise in our collision-oblivious designs.

II. The Oligo Design Problem

In this section, we formally define the problem of oligo design for gene synthesis, along with some useful notation. A *DNA strand*, or *oligo*, is a string over alphabet $\{A, C, G, T\}$. We consider the left and right ends of the string to represent the $5'$ and $3'$ ends, respectively, of the corresponding physical DNA strand. A *DNA duplex* S consists of two complementary DNA strands distinguishable by a value σ; S^σ refers to the sense strand when $\sigma = 0$ and the anti-sense strand when $\sigma = 1$. The complement $S^{1-\sigma}$ of DNA strand S^σ is obtained from S^σ by replacing each A with a T and vice versa, each C with a G and vice versa, and reversing the resulting string. Thus, for example, the *complement* of a strand having a sequence AATGGG is CCCATT. In this manner, for some sense strand $S^0 = s_1 s_2 \dots s_{|S|}$, we let the oligo $O^0_{a,b}$ denote the substring $s_a \dots s_b$ and $O^1_{a,b}$ denote the substring $\overline{s_a \dots s_b}$, the complement of $s_a \dots s_b$.

We next formally define what is an oligo design. See Fig. 1 for an example.

Definition 2.1: For a fixed DNA duplex S, a *gapped oligo design* $\mathcal{O}$ is specified by two strictly monotonically increasing sequences of indices: $i_1, i'_1, i_2, i'_2, \dots, i_x, i'_x$ and $j_1, j'_1, j_2, j'_2, \dots, j_y, j'_y$, such that:

- $|x - y| \leq 1$,
- $\min(i_1, j_1) = 1$ and $\max(i_x, j_y) = |S|$,
- the indices from the sequences alternately overlap in the following manner:
  - if $i_1 = 1$ then $x \geq y$ and
    - $i_k < j_k \leq i'_k < j'_k$, $1 \leq k \leq y$
    - $j_y < i_x \leq j'_y < i'_x$, if $x > y$
  - if $j_1 = 1$ then $y \geq x$ and
    - $j_k < i_k \leq j'_k < i'_k$, $1 \leq k \leq x$
    - $i_x < j_y \leq i'_x < j'_y$, if $y > x$

The *set of oligos corresponding to* $\mathcal{O}$ is

$$\begin{aligned}\text{set}(\mathcal{O}) = &\{O^0_{i_k,i'_k} = s_{i_k} \dots s_{i'_k} \mid 1 \leq k \leq x\} \\ &\cup \{O^1_{j_k,j'_k} = \overline{s_{j_k} \dots s_{j'_k}} \mid 1 \leq k \leq y\}.\end{aligned}$$

An *ungapped oligo design* is a restricted version of a gapped oligo design where $i_k = i'_{k-1} + 1$, $1 < k \leq x$, and $j_k = j'_{k-1} + 1$, $1 < k \leq y$.

The remaining definitions are with respect to a fixed DNA duplex S and oligo design $\mathcal{O}$ for S.

Definition 2.2: A *designed hybridization* is a pair of oligos $O^\sigma_{a,b}, O^{1-\sigma}_{c,d} \in \text{set}(\mathcal{O})$ where either $a < c \leq b < d$ or $c < a \leq d < b$. These oligos share a *complementary overlap region.* (See Fig. 1)

Let $\text{Tm}(O, O')$ and $\text{Tm}(O)$ be the melting temperatures of the MFE duplex secondary structure formed by oligos O and O', and of the secondary structure formed by O alone, respectively. Let $\text{Tm}(O^\sigma_{a,b} \cap O^{1-\sigma}_{c,d})$ be the melting temperature of the complementary overlap region associated with a designed hybridization between $O^\sigma_{a,b}$ and $O^{1-\sigma}_{c,d}$.

Finally, for any oligo $O \in \text{set}(\mathcal{O})$, let $T_{min}(O)$ and $T_{min}(\mathcal{O})$ be the minimum of all the melting temperatures of designed hybridizations involving O, and overall in $\mathcal{O}$, respectively.

Definition 2.3: An oligo design $\mathcal{O}$ is:

- *length range limited* if all oligos in set($\mathcal{O}$) have length in the range $[l_{min}, l_{max}]$
- *structure free* if for all oligos $O \in \text{set}(\mathcal{O})$, $\text{Tm}(O) \leq t_{sh}$, the threshold for self hybridization
- *designed Tm satisfied* if for all oligo pairs $(O^\sigma_{a,b}, O^{1-\sigma}_{c,d}) \in \mathcal{O}$ having a designed hybridization, $t_{min} \leq \text{Tm}(O^\sigma_{a,b} \cap O^{1-\sigma}_{c,d}) \leq t_{max}$, where t_{min} and t_{max} are the threshold of the minimum and maximum overlap region melting temperature, respectively

Definition 2.4: Given length range $[l_{min}, l_{max}]$ and temperatures t_{sh}, t_{min} and t_{max}, $\mathcal{O}$ is *valid* if and only if it is length range limited, structure free and designed Tm satisfied. Let $\mathcal{D}(S)$ be the set of all valid oligo designs for S.

Definition 2.5: Given threshold t_{col}, $\mathcal{O}$ is *collision free* if, for any pair of oligos (O, O') that do not have a designed hybridization, $\text{Tm}(O, O') + t_{col} \leq \min(T_{min}(O), T_{min}(O'))$. If $\text{Tm}(O, O') + t_{col} > \min(T_{min}(O), T_{min}(O'))$, we say that (O, O') is an *oligo collision.*

An alternative definition is that for any pair of oligos (O, O') that do not have a designed hybridization, $\text{Tm}(O, O') + t_{col} \leq T_{min}(\mathcal{O})$. Our method for collision aware synthon design in Sect. IV also applies to this definition.

We note that the above definitions can be expanded to consider additional design criteria. For instance, dependent on the synthesis protocol, it may be necessary to ensure all oligos in a design can be easily amplified by PCR. Although we do not consider this design property further, it and other criteria imposed on individual oligos could be modeled in a similar manner as those presented in this study.

Informally, a valid design is a requirement of any potential oligo design solution. Therefore, given that the constraints must be satisfied, we may choose to define some objective score function, g, of oligo designs, to optimize one or more design criteria. For example, in the remainder of this study, we have chosen to define $g(\mathcal{O}) = t_{max} - T_{min}(\mathcal{O})$ in order to minimize the range of melting temperatures of designed overlap regions. We could have similarly defined $g(\mathcal{O}) = T_{max}(\mathcal{O}) - t_{min}$ to achieve a similar effect. However, the definition of g is not restricted to design criteria alone. Consider the case where there is a known cost per nucleotide in the gene synthesis process. Then, of all possible valid designs, it may be beneficial to define g such that it would choose the design which requires the least number of nucleotides to be synthesized.

We now formally define the collision oblivious and collision aware oligo design problems.

Collision Oblivious Oligo Design for Gene Synthesis (CO-ODGS)

Instance: DNA duplex S, oligo length range $[l_{min}, l_{max}]$, maximum melting temperature of self-hybridization, t_{sh}, and minimum and maximum overlap region melting temperatures, t_{min} and t_{max}.

Problem: Find a valid oligo design $\mathcal{O}^*$ for S, such that $g(\mathcal{O}^*) = \min\{g(\mathcal{O}')|\mathcal{O}' \in \mathcal{D}(S)\}$ where g is some objective score function of oligo designs. $\mathcal{O}^*$ is an *optimal design with respect to g.*

Collision Aware Oligo Design for Gene Synthesis (CA-ODGS)

The collision aware oligo design for gene synthesis problem (CA-ODGS) is defined as the CO-ODGS problem with the added constraint that the design must also be collision free.

III. Collision-Oblivious Algorithms

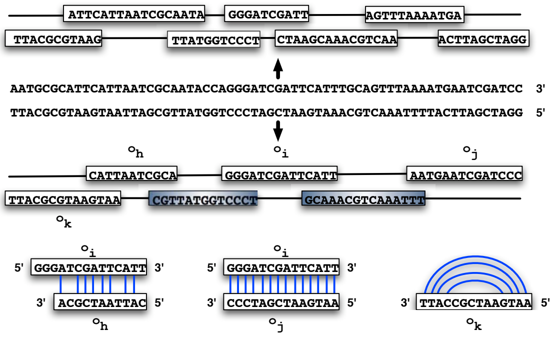

Fig. 1. An input sequence with a potentially valid design (top) and an invalid design (below), having a potential oligo collision (bottom left), a perfect oligo collision (bottom center), and an oligo likely to self-hybridize (bottom right). Shaded oligos have a designed hybridization with o_i.

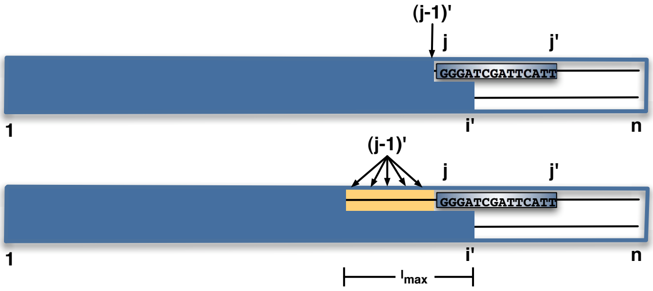

Fig. 2. The ungapped recurrence (top) attempts to find the best starting position j for an oligo ending at j', given that there is an oligo ending at i' on the opposite strand. Position $(j-1)'$ immediately precedes j in an ungapped design. The gapped recurrence (bottom) also considers the best position for the oligo ending at $(j-1)'$, given that it overlaps the oligo ending at position i'.

We now describe two dynamic programming algorithms that are guaranteed to find the optimal design with respect to the objective function g, if one exists, for any instance of the CO-ODGS problem detailed in Sect. II. We start by describing an algorithm for ungapped oligo designs and show how the algorithm can be generalized to handle the case of gapped designs.

A. Ungapped Oligo Designs

In Eqn. (1), $D^{\sigma}_{i',j'}$ determines the score of the optimal design with respect to function g', having an oligo ending at position j', on strand σ, and another ending at position i' on the opposite strand (see Fig. 2). Intuitively, the recurrence evaluates all possible values of j, denoting the oligo $O^{\sigma}_{j,j'}$, for 1) constraint satisfaction, and 2) optimization with respect to the scoring function g'. Oligo $O^{\sigma}_{j,j'}$ is first checked to ensure it is structure free with respect to the self hybridization melting temperature threshold t_{sh} (line 1). The base case is reached when $j = 1$ and the oligo is structure free (line 2). In the recursive case, $j \neq 1$, the new overlap region introduced $(s_j \dots s_{i'})$ is evaluated to ensure it is designed Tm satisfied (lines 3,4). If all constraints have been satisfied, then the score is evaluated and defined to be the larger of $D^{1-\sigma}_{j-1,i'}$, the score of the optimal design with an oligo ending at position $j - 1$

on strand σ and one ending at position i on strand $1-\sigma$, and of the new score associated with oligo $O^{\sigma}_{j,j'}$, evaluated by g'. $D^{\sigma}_{i',j'} = \min_{\max(j'-l_{max}+1,1)\leq j\leq \min(j'-l_{min}+1,i')}$

$$\left\{\begin{array}{ll} \infty & \text{, if } \mathrm{Tm}(O^{\sigma}_{j,j'}) > t_{sh} \\ 0 & \text{, if } j=1 \wedge \mathrm{Tm}(O^{\sigma}_{j,j'}) \leq t_{sh} \\ \infty & \text{, if } \begin{array}{l} j \neq 1 \wedge \\ \mathrm{Tm}(O^{\sigma}_{j,i'} \cap O^{1-\sigma}_{j,i'}) > t_{max} \end{array} \\ \infty & \text{, if } \begin{array}{l} j \neq 1 \wedge \\ \mathrm{Tm}(O^{\sigma}_{j,i'} \cap O^{1-\sigma}_{j,i'}) < t_{min} \end{array} \\ \begin{array}{l} \max(D^{1-\sigma}_{j-1,i'}, \\ g'(j,i')) \end{array} & \text{, otherwise} \end{array}\right\} \tag{1}$$

$$g'(j,i') = t_{max} - \mathrm{Tm}(O^{0}_{j,i'} \cap O^{1}_{j,i'}) \tag{2}$$

$$D^{*}_{j'} = \min_{j'-l_{max}<i'<j'} \left\{ \min_{0\leq\sigma\leq 1} \left\{ D^{\sigma}_{i',j'} \right\} \right\} \tag{3}$$

In this particular formulation, we chose to optimize the range of designed overlap melting temperatures by defining our objective optimization function $g(\mathcal{O}) = t_{max} - T_{min}(\mathcal{O})$. In order to determine the effect to the score when choosing a particular oligo, $O^{\sigma}_{j,j'}$, we have defined a function g' with respect g (see Eqn. (2)). We reiterate here that g could be defined to optimize for another design criteria if desired. Likewise, additional constraints could be imposed and existing ones removed, dependent on the application. Furthermore, while we have defined each hybridization related constraint in terms of melting temperature these could easily be expressed in terms of Gibbs free energy change.

In Eqn. (3), $D^{*}_{j'}$ defines the optimal score of a design of the prefix of S of length j'. With respect to j', it evaluates all possible placements of an oligo ending at some position i' on the opposite strand. Therefore, for some sequence S, having length $|S|$, the optimal design score is $D^{*}_{|S|}$.

B. Gapped Oligo Designs

Unlike the special case of ungapped designs where $(j-1)' = j-1$, in a gapped design the previous oligo on strand σ could end at a number of valid positions, denoted as position $(j-1)'$, given that an oligo covers position $(j-1)'$ on strand $1-\sigma$. Therefore, to generalize Eqn. (1) for gapped designs, replace $D^{1-\sigma}_{j-1,i'}$ in the recursive case, with:

$$\min_{i'-l_{max}<(j-1)'<j} \left\{ D^{1-\sigma}_{(j-1)',i'} \right\} \tag{4}$$

C. Time and Space Complexity

We assume that satisfaction of all design constraints can be calculated in constant time (which depends on l_{min} and l_{max}). Let S be the DNA duplex of the problem instance and $n = |S|$. In the case of the ungapped algorithm, for each possible pair $(i',j') \in \{(i',j') \mid 1 < i',j' \leq |S| \wedge 1 \leq |j-i| < l_{max}\}$, every possible j must be evaluated to determine the score contributed by oligo $O^{\sigma}_{j,j'}$. There are at most $l_{max} - l_{min} + 1$ possible placements of j, given any (i',j'). Therefore, the ungapped algorithm runs in time $O((l_{max} - l_{min}) \cdot l_{max} \cdot n) = O(n)$, as l_{min} and l_{max} are design constants. An entry must be stored in the dynamic programming table for every (i',j'), $1 \leq |i'-j'| < l_{max}$, for both strands, therefore $O(l_{max} \cdot n) = O(n)$ space is needed. If only a score is required, the space can be reduced to $O(1)$.

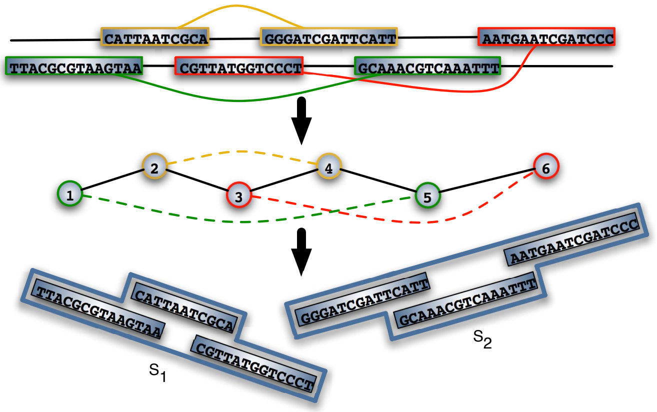

Fig. 3. An oligo design with collisions denoted by edges (top) is transformed into an oligo collision graph (middle) with designed hybridizations shown with solid edges and collisions shown with dashed edges. A minimum-size partition of the graph, representing synthons, is shown (bottom).

For the gapped design algorithm, the space remains the same. As there are at most l_{max} possible placements of $(j-1)'$, in contrast to one valid position in the ungapped case, the time complexity is $O(l_{max} \cdot (l_{max} - l_{min}) \cdot l_{max} \cdot n) = O(n)$.

IV. Collision Aware Design Using Synthons

We conjecture that the CA-ODGS problem is $\mathcal{NP}$-hard. We have previously presented evidence of this by showing an abstraction of the problem, collision aware string partition (CA-SP), is $\mathcal{NP}$-complete [11]. Given that a polynomial time algorithm for the CA-ODGS problem is unlikely, we adopt a heuristic approach.

Informally speaking, the heuristic we employ does not attempt to minimize the number of oligo collisions. Rather, it partitions an optimal collision oblivious design $\mathcal{O}$ into a minimum number of synthon regions which are collision free. We note that there may exist another optimal collision oblivious design which produces fewer synthons, however, our algorithm makes no attempt to find such a design. As we demonstrate in Sect. V this simple approach is very effective in practice.

First, given the optimal valid design $\mathcal{O}$ for an input duplex S, construct the *oligo collision graph* $G = (V, E = E' \cap E'')$, in which $V = \mathrm{set}(\mathcal{O})$ and each node is labeled according to the oligo order. That is, the oligo which covers the first position of the duplex is labeled '1', then the first oligo on the opposite strand is labeled '2', and the remaining labels are assigned by alternating between strands from left to right relative to the 5' end of the sense strand. E' is the set of edges representing designed hybridizations, and E'' the edges representing oligo collisions, based on our previous definitions. Refer to Fig. 3 for an example with designed hybridizations shown with solid edges and collisions shown with dashed edges.

Second, given G, determine the minimum-size partition P of V such that $G[p]$ is connected and does not contain an edge from E'', $\forall p \in P$. A minimum-size partition of the collision graph corresponds to the minimum number of collision free synthons required to cover $\mathcal{O}$. In Sect. IV-A, we present a dynamic programming algorithm for this task.

A. Synthon Partition Algorithm

Intuitively, D'_k from Eqn. (5) is the minimum number of collision free partitions (synthons) of an oligo collision graph G required to cover the collision oblivious design $\mathcal{O}$, up to oligo k. The recurrence determines the best start position i of a synthon ending at position k. For a potential synthon consisting of oligos $i, \ldots, k$, if there is a collision between any pair of these oligos, then the synthon is considered invalid (line 1). The base case occurs when a synthon begins at the first oligo and no pair of oligos in the proposed synthon are in conflict (line 2). Otherwise, the recursive case adds an additional synthon to the score of the previous best solution at D'_{i-1} (line 3). Therefore, the minimum number of collision free synthons required to cover set($\mathcal{O}$) having x oligos is given by D'_x. We set $D'_0 = 0$, and for $1 \leq k \leq x$ we have

$$D'_k = \min_{1 \leq i \leq k} \left\{ \begin{array}{ll} \infty & \text{, if } \exists j, i \leq j \leq k, (j,k) \in E'' \\ 1 & \text{, if } \forall j, i \leq j \leq k, (j,k) \notin E'' \wedge (i = 1) \\ D'_{i-1} + 1 & \text{, otherwise} \end{array} \right\} \tag{5}$$

B. Time and Space Complexity

We assume that an oligo collision graph is given as input to the synthon partition algorithm. However, we note this graph can be constructed naively in $O(x^2)$ time by comparing every pair of oligos under the assumption that the collision condition can be calculated in some constant time.

The synthon partition algorithm is quadratic in the number of oligos, x, in $\mathcal{O}$. Since for every ending position k, $1 \leq k \leq x$, all possible starting positions of the synthon must be evaluated, $1 \leq i \leq k$, then in the worst case, $O(x^2)$ time is required. As we must store an entry in the dynamic programming table for each k, $1 \leq k \leq x$, then $O(x)$ space is required.

V. Experimental Analysis

We divide our analysis into two main sections. In Sect. V-B we ask the following questions. First, for various values of t_{max}, how effective is the gapped and ungapped collision-oblivious algorithm in finding valid designs, not necessarily collision free, on real data? Second, of the valid designs found by both the gapped and ungapped algorithms, to what degree do collisions arise? In Sect. V-C we run the gapped collision-oblivious implementation on a larger number of sequences and for each valid design that results, we ask the question: what is the minimum number of synthons required to make the design collision free? Finally, we report on the runtime performance of the algorithms. Following a short description of our data set and implementation details, we report the results of our analyses in the rest of this section.

A. Experimental Environment

1) Data Set: We use random samples of the 5,629 CDS (coding DNA sequence) regions of the ENCODE dataset [12] (version hg17 NCBI build 35). This curated dataset comprises approximately 1% of the human genome and is representative of several its characteristics such as distribution of gene lengths and GC composition (54.31%). The CDS regions range in length from 85 to 8185 bases, averaging 172 bases with 267 bases standard deviation.

2) Implementation and Hardware: In all experiments, we fix the oligo length range $[l_{min}, l_{max}] = [37, 52]$ and set $t_{sh} = t_{min} = 45$°C. The value of t_{max} varies and details are given for each experiment set. We set the threshold for oligo collisions to be $t_{col} = 5$°C. Calculation of melting temperature values, denoted by the function Tm in Sect. II, were performed by the PairFold (for duplexes) and SimFold (for single strands) structure prediction software of Andronescu *et. al* [13]. All algorithms were implemented in C++ and compiled with g++ (GCC 3.3). Experiments were run on our reference Pentium IV 2.4 GHz processor machines, with 1GB main memory and 256 Kb of CPU cache, running SuSE Linux version 9.1.

B. Oligo Designs in Practice

To better understand the practical effectiveness of the collision-oblivious algorithm in finding valid solutions and the number of collisions which occur in these designs, we first conducted a set of experiments on a limited dataset of one hundred sequences chosen uniformly at random from our reference dataset. For each of the one hundred sequences, both the ungapped and gapped collision-oblivious algorithms were run for twelve different values of $t_{max} = \{35, 40, 45, \ldots, 90\}$.

1) Efficacy of the Gapped and Ungapped Versions: The gapped version of the algorithm is much more successful in finding valid designs (see Fig. 4 left side). This version finds a valid design for all one hundred sequences, regardless of the value of t_{max}. In contrast, the ungapped version is unable to find a valid design for any sequence when $t_{max} <$ 70°C. Even for the highest value of t_{max} we tested, 90°C, there was still one sequence where an ungapped design was impossible, given the design constraints. Overall, of the 1200 design attempts of the ungapped algorithm, only 234 valid designs were found. Clearly, the added design flexibility of the gapped version is crucial for designs requiring a low or moderate value for t_{max}.

2) Frequency of Oligo Collisions: For each valid design found, PairFold was used to determine the number of oligo collisions based on the protocol previously described. In Fig. 4 (right side), we report the number of collisions per 100 bases for each valid design found, both gapped and ungapped, in an attempt to normalize for sequence length. All valid designs are grouped by t_{max} and a lowess regression curve of best fit has been added. Although there is a clear inverse relationship between the collision rate and t_{max}, even at moderate design temperatures this rate is very low. When $t_{max} = 50$°C, there are less than two collisions per 100 bases on average. This suggests it might be possible to adapt the collision-oblivious algorithm to eliminate collisions.

C. Synthon Design and Algorithm Performance

Motivated by evidence that the rate of oligo collisions is low for moderate design temperature, we conducted an extended study of five hundred sequences chosen uniformly at random from our reference dataset. Due to the poor design success rate of the ungapped algorithm, we omit it from further analysis. For each of the five hundred sequences, the gapped collision-oblivious design algorithm was run for each value of $t_{max} = \{50, 60, 70, 80\}$. For each valid design which resulted, the collision conflict graph was constructed based on the protocol previously described. The synthon design algorithm was then run to determine the minimum number of synthons needed to have a collision free partition of the original valid design. Runtime statistics were tracked throughout and are discussed below.

1) Minimum Required Synthons: In Fig. 5 (left side), the cumulative distributions of required synthons over the five hundred sequences is plotted for each value of t_{max}. The worst case occurs for $t_{max} = 50$°C when 21 synthons are required to partition a 2.4kb sequence into collision free regions. However, even at this lowest temperature tested, approximately 80% of all sequences require two synthons or less to become collision free. Each successive value of t_{max} further improves upon this result with $t_{max} = 70$°C requiring at most 3 synthons (for 3 of 500 sequences) and $t_{max} = 80$°C requiring at most two synthons for any sequence.

2) Runtime Performance: In Fig. 5 (right side), the CPU runtime of the collision-oblivious design algorithm is plotted for each of the five hundred sequences when $t_{max} = 70$°C. The worst case is a design time of approximately 4.5 seconds for a sequence 8.4kb in length. All other values of t_{max} result in the same characteristic performance and differ only within a small constant factor. For the synthon design algorithm, an overall worst case running time of 0.02 seconds is reported for the same 8.4kb sequence.

We stress that the algorithms detailed in this study are invariant to those used for detecting self-hybridization, collisions and calculating melting temperature of designed overlap regions; however, for completeness we summarize the performance of SimFold and PairFold which we employed for this task. In all cases, self-hybridization, using SimFold, and melting temperature of designed overlap regions, using PairFold, were pre-computed for every possible oligo and every possible overlap region. Calculation of Tm values of overlap regions ranged in runtime from 0.2 to 14.8 CPU seconds, with a mean runtime of 2.3 seconds having standard deviation 2.0 seconds. Runtime required to precompute self-hybridization ranged from 14.1 to 1225.8 CPU seconds with a mean runtime of 190.1 seconds having a standard deviation of 167.8 seconds. PairFold was also used to derive the conflict graphs for the synthon design algorithm. For this task, CPU runtime ranged from 0.4 to 3997.4 seconds with a mean runtime of 151.1 seconds having standard deviation 336.7 seconds.

VI. Conclusions

In this paper, we presented an efficient dynamic programming algorithm for ungapped and gapped collision oblivious oligo design for gene synthesis and showed the gapped variant to be highly effective in practice. We provided empirical evidence, using a large gene set, that oligo collisions occur infrequently in the designs produced by our collision oblivious algorithm. Motivated by this fact and previous evidence suggesting the collision aware oligo design problem is $\mathcal{NP}$-hard, we described and evaluated an efficient synthon partition algorithm which determines the minimal number of regions required to make a design collision free. We have shown that for reasonable parameters, two synthons are usually sufficient to achieve this design goal.

Future work includes the expansion of the oligo design program to incorporate codon bias optimization, a common feature of available gene design software packages. Additional study of ungapped oligo designs is also warranted in order to improve design success, possibly under less stringent design criteria. This study has focused on designs of coding regions only. However, design efficacy should also be evaluated for non-coding sequences, such as promoter regions, which are likely to require synthesis in some applications. Finally, despite motivating evidence suggesting collision aware oligo design is $\mathcal{NP}$-hard, the question still remains open.

References

[1] J. C. Cox, J. Lape, M. A. Sayed, and H. W. Hellinga, "Protein fabrication automation," *Protein Sci*, vol. 16, no. 3, pp. 379–390, Mar 2007.

[2] D. M. Hoover and J. Lubkowski, "DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis," *Nuc Acids Res*, vol. 30, no. 10, May 2002.

[3] W. P. Stemmer, A. Crameri, K. D. Ha, T. M. Brennan, and H. L. Heyneker, "Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides," *Gene*, vol. 164, no. 1, pp. 49–53, Oct 1995.

[4] J. Tian, H. Gong, N. Sheng, X. Zhou, E. Gulari, X. Gao, and G. Church, "Accurate multiplex gene synthesis from programmable DNA microchips," *Nature*, vol. 432, no. 7020, pp. 1050–1054, Dec 2004.

[5] S. Jayaraj, R. Reid, and D. V. Santi, "GeMS: an advanced software package for designing synthetic genes," *Nuc Acids Res*, vol. 33, no. 9, pp. 3011–3016, 2005.

[6] J. M. Rouillard, W. Lee, G. Truan, X. Gao, X. Zhou, and E. Gulari, "Gene2Oligo: oligonucleotide design for in vitro gene synthesis," *Nuc Acids Res*, vol. 32, no. Web Server issue, pp. 176–180, Jul 2004.

[7] R. Rydzanicz, X. S. Zhao, and P. E. Johnson, "Assembly PCR oligo maker: a tool for designing oligodeoxynucleotides for constructing long DNA molecules for RNA production," *Nucl Acids Res*, vol. 33, no. Web Server issue, pp. 521–525, Jul 2005.

[8] A. Villalobos, J. E. Ness, C. Gustafsson, J. Minshull, and S. Govindarajan, "Gene Designer: a synthetic biology tool for constructing artificial DNA segments," *BMC Bioinformatics*, vol. 7, pp. 285–285, 2006.

[9] G. Wu, N. Bashir-Bello, and S. J. Freeland, "The synthetic gene designer: a flexible web platform to explore sequence manipulation for heterologous expression," *Protein Expr Purif*, vol. 47, no. 2, pp. 441–445, Jun 2006.

[10] D. Zha, A. Eipper, and M. T. Reetz, "Assembly of designed oligonucleotides as an efficient method for gene recombination: a new tool in directed evolution," *Chembiochem*, vol. 4, no. 1, pp. 34–39, Jan 2003.

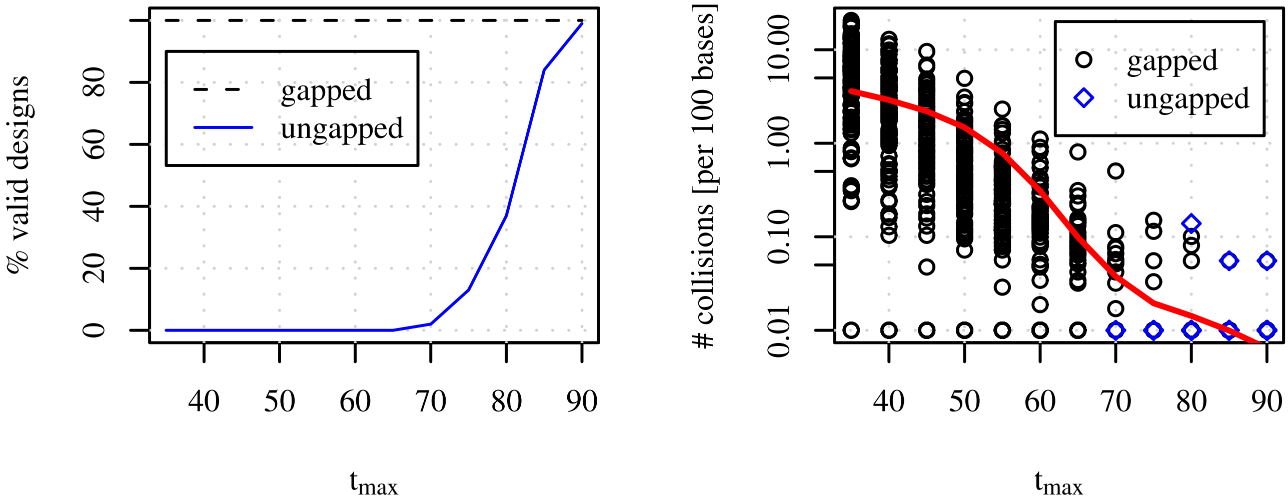

Fig. 4. The gapped algorithm is much more successful at finding valid designs (left side). For valid designs of both variants, oligo collisions occur infrequently (right side).

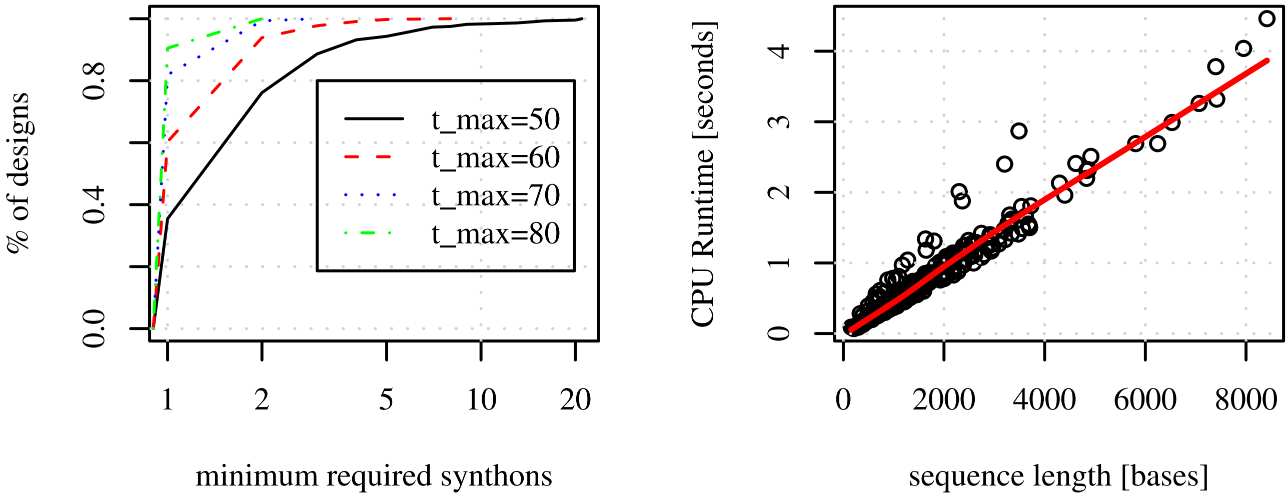

Fig. 5. For moderate values of t_{max}, very few synthons are required to make a design collision free (left side). Run time performance of the collision-oblivious algorithm, which scales linearly, is plotted for 500 designs of genes of varying lengths (right side).

[11] A. Condon, J. Maňuch, and C. Thachuk, “On complexity of collision-aware oligo design for gene synthesis,” Submitted.

[12] The ENCODE Consortium, “The ENCODE (ENCyclopedia Of DNA Elements) project,” *Science*, vol. 306, no. 5696, pp. 636–640, Oct 2004.

[13] M. Andronescu, Z. C. Zhang, and A. Condon, “Secondary structure prediction of interacting RNA molecules,” *J Mol Biol*, vol. 345, no. 5, pp. 987–1001, Feb 2005.