Title

Novel algorithms for the *in vitro* gene synthesis oligo design problem

Presenter

Chris Thachuk[1]

Additional Co-Authors

Anne Condon[2] and Arvind Gupta[1]

Abstract

Massive amounts of biological information is now commonplace. One use of this knowledge is the creation of synthetic genetic sequences for a variety of applications. These include over-expression of the corresponding gene product, introduction of the sequence into a biological system and investigation of regulatory pathways amongst other biological and potential industrial uses.

In the oligo design problem for *in vitro* gene synthesis, we wish to design a feasible set of short oligonucleotides which can be assembled to some genetic sequence. The sequence need not be naturally occurring as each oligo is ultimately synthesized within a laboratory. The input for this problem is a string S denoting a desired double stranded DNA sequence, comprised of a sense and anti-sense strand. The problem is to disassemble S into fragments, or oligos, such that it is feasible to efficiently synthesize the oligos (via parallel chip synthesis methods), to amplify them (via PCR), and to assemble them (via hybridization) in the correct order, thus creating multiple identical copies of S. The assembly may be performed in two stages, by first forming subassemblies, which are then assembled into the full sequence S. All of these requirements impose multiple constraints on a feasible solution (set of oligos): oligos should be short; the range of melting temperature for overlap regions of two oligos (on the sense and anti-sense strands of S) should be narrow, and there should be low likelihood of oligo self-hybridization (such as formation of a hairpin loop) or of mis-hybridization of two oligo in the set, an event we refer to as *oligo collision*.

In this work, we present dynamic programming solutions to the above problem when gaps between adjacent oligos on the same strand are not permitted (ungapped algorithms) and for the case of permitting gaps (gapped algorithms). We first ignore the constraint of oligo collision. We adopt a two stage approach, corresponding to a two-stage assembly process. In the first stage, we consider every possible design region of size L within S, with $L_l \leq L \leq L_u$ and L_l and L_u are user-defined parameters. Our algorithm determines a feasible solution (if one exists) for each possible design region, in worst case time $O(L)$ per region, assuming that the length of oligos and the melting temperature overlap range are both bounded by a constant. In the second stage, our algorithm determines, in time linear in $|S|$, a sequence of design regions with feasible solutions which cover the whole input S, if such a sequence exists.

In ongoing work, we are extending our approach to consider oligo collision, a constraint which greatly increases the difficulty of the design task. In this version, every pair of oligos within each design region must be checked for possible collision. We can bound the time complexity of finding a feasible solution for a design region (if one exists) as a function of four quantities: L, the length of the design region; l, a user-specified lower bound on the length of an oligo; $|\tau|$, the size of the *tile set* τ for the region, where a tile is formed by two partially hybridized overlapping oligos; and C, where $\Theta(C)$ is the complexity of determining which oligos in the design region collide. The time complexity bound is $O(|\tau|^{\lfloor L/l \rfloor} C)$. To ensure practical runtimes, we are investigating the use of efficient string matching techniques and data structures, including generalized suffix trees, to filter out oligos in a design region which possibly collide. Pairs of oligos which pass the filter are then checked using secondary structure prediction algorithms. In future work, we will consider addition of primers and restriction sites to oligos, and codon optimization for potential host organisms.

[1]Simon Fraser University
[2]University of British Columbia