Greedy Algorithms for Optimized DNA Sequencing

Allon G. Percus[*] David C. Torney[†]

Abstract

We discuss the problem of optimally “finishing” a partially sequenced segment of DNA strand. While a simplified model may be solved to optimality by a greedy algorithm, adding more realistic features appears to make the problem computationally hard. We propose an Integer Linear Programming formulation that places it in the context of more general coverage problems, in the hope that this will provide further insight into the solvability of this biologically important application.

1 Background

DNA sequencing experiments are typically performed in two stages: *shotgun sequencing* and *walking*. Shotgun sequencing may be thought of as a stochastic process, where many short subintervals at random locations on a DNA strand are sequenced. A cost C_S is associated with each subinterval sequenced. Walking is a deterministic finishing process, where regions insufficiently covered in the shotgun process may be sequenced. A higher cost C_W is associated with each subinterval sequenced in this way. Note that both procedures sequence discrete subintervals, which may all be taken to have equal (unit) length.

Current standards in the Human Genome Project require every position on the DNA strand to be sequenced at least 3 times to insure minimum reliability of results. Any moderate amount of shotgun sequencing will tend to leave *some* regions of the DNA strand insufficiently sequenced according to this criterion; once the locations of these regions have been established, they must be finished by walking. Since shotgun positions are random, these regions are generally of non-integer length, whereas the walking procedure sequences one unit length at a time. It is then a non-trivial problem to decide exactly *where* to place the walks in order to meet the required criterion with minimal redundancy. Furthermore, in spite of its importance to the genome project’s ambitious timetable, this problem does not appear to have been addressed yet in the literature.

2 Simplified Model

Consider a typical instance of shotgun coverage, as depicted in Figure 1. A single walk, from any starting position, has the effect of increasing the profile height by one, over one unit length. The goal is to fill the deficient regions in the profile, *i.e.*, the areas with height less than 3, using as few walks as possible. A simple greedy algorithm for doing this might go as follows: proceed from one end of the profile to the other (left-to-right or right-to-left), and whenever a deficiency is found, perform a walk as many times as necessary *at that position* to eliminate the deficiency. The profile height will thus be increased, over one unit length, by the number of times a walk is performed there. Interestingly, it may be shown that this greedy algorithm — as well as other variations on it — gives an optimal solution.

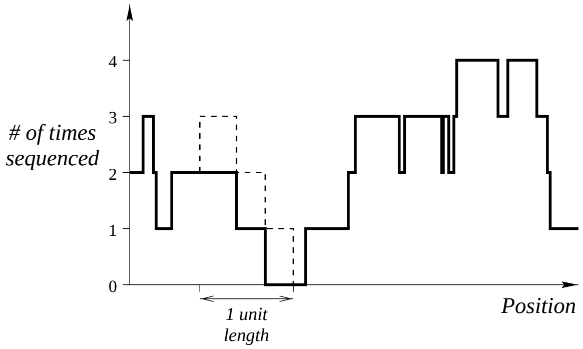

Figure 1: Coverage profile, showing the number of times positions along the DNA strand have been sequenced in shotgun sequencing. Dashed line shows new profile height after one walk.

3 General Model

Now modify the original statement of the problem so as to make it more faithful to experimental reality, though also more complex. When walks are performed multiple times at a given position, let only the first walk cost

[*] CIC–3 and Center for Nonlinear Studies, MS–B258, Los Alamos National Laboratory, Los Alamos, NM 87545. E-mail: percus@lanl.gov.

[†] T–10 and Center for Nonlinear Studies, MS–K710, Los Alamos National Laboratory, Los Alamos, NM 87545. E-mail: dct@lanl.gov.

C_W; all subsequent walks at that position may take advantage of laboratory material already prepared for the first one, and have a marginal cost that is in fact close to C_S. A p-fold multiple walk (increasing the profile by height p over a unit length) may now be taken to have cost $C_W + (p-1)C_S$, for $p \geq 1$. It might thus be profitable to perform multiple walks over a given region, even when this results in a greater total number of walks than would otherwise be necessary.

Consider the problem in the following terms. Define $\alpha = (C_W - C_S)/C_S$, so that the cost of a p-fold multiple walk is $C_S(p + \alpha)$ for $p \geq 1$. When $p = 0$, however, the cost is 0, so the presence of $\alpha \neq 0$ in the general model introduces a non-linearity. The generic greedy algorithms used on the simplified model are no longer optimal: the best we can say is that they give, in linear time, a solution whose cost is within a constant factor $(1 + \alpha)$ times the optimum. One could also propose various greedy-based heuristics to find near-optimal solutions, such as scanning from one end of the profile to the other, and over each unit-length interval performing a p-fold multiple walk, where p is the greatest deficiency found anywhere in that interval. A more structured approach, though, is to rephrase the non-linear problem so that it can in fact be given by an Integer Linear Programming formulation.

4 ILP Formulation

The first observation to make is that there is only a finite number of discrete locations at which the initial coverage profile changes, equal to the number of shotgun subintervals used in the instance. As all walks are of one unit length, it is sufficient to consider positions on the strand that correspond to these locations plus or minus integer values. We are given:

$$\begin{aligned} \alpha &= \text{ratio of fixed to incremental walking cost} \\ m &= \text{length of DNA strand} \\ n &= \text{\# of shotgun subintervals (of unit length)} \\ d_i &= \text{deficiency at position } i,\ i = 1, \ldots, mn \end{aligned}$$

Now consider all subintervals that could be used in walking, *i.e.*, subintervals of unit length beginning at position i, $i = 1, \ldots, mn$. Each of these subintervals represents the set of positions $[i, i+n)$. Generate 3 identical copies of each set, and define:

$$\begin{aligned} S_1, S_2, S_3 &= \text{subintervals beginning at position 1} \\ S_4, S_5, S_6 &= \text{subintervals beginning at position 2} \end{aligned}$$

and so on, up to S_N (where $N = 3mn$):

$$S_{N-2}, S_{N-1}, S_N = \begin{array}{l}\text{subintervals beginning at} \\ \text{position } mn\end{array}$$

Define a weight w_i for each subinterval S_i, such that:

$$\begin{gathered} w_1 = 1, w_2 = 2, w_3 = 3 \\ w_4 = 1, w_5 = 2, w_6 = 3 \\ \vdots \\ w_{N-2} = 1, w_{N-1} = 2, w_N = 3. \end{gathered}$$

Let subinterval S_i with weight w_i correspond to a w_i-fold multiple walk at position i. Let $x_i = 1$ if such a walk is performed, and $x_i = 0$ if it is not.

The optimization problem then reduces to the following canonical form ILP. Minimize

$$\sum_{i=1}^{N} (\alpha + w_i) x_i \tag{4.1}$$

subject to the constraint that $\forall j$,

$$\sum_{i:\ j \in S_i} w_i x_i \geq d_j \qquad \text{and} \qquad x_j \in \{0, 1\}. \tag{4.2}$$

Note that while in principle this allows us to choose more than one copy of a subinterval at a given position, the minimization of (4.1) will necessarily favor instead a single copy with equivalently higher weight.

5 Discussion

Efficient DNA sequencing is an important problem in biology. We have seen how simple machinery from combinatorial optimization can profitably be adapted to this application. While the general model we have discussed ($\alpha > 0$) *appears* to be computationally hard, it is not at present certain whether this is really the case. The question of how the nature of an optimal algorithm changes with increasing α remains open. The ILP formulation introduces conceptual simplicity to the problem, however, and makes us more hopeful that its computational complexity can be determined. Finally, it provides a framework for approaching the problem when modified to take account of other experimental realities, such as the double-stranded nature of DNA and a requirement that *both* of these strands be sequenced at least once at each position.