The Simplified Partial Digest Problem: Hardness and a Probabilistic Analysis

Zoë Abrams[1] and Ho-Lin Chen[2]
Stanford University Computer Science Department
{zoea, holin}@stanford.edu

Introduction

We study the problem of genome mapping using *restriction site analysis*. In restriction site analysis, an enzyme cuts a target DNA strand into DNA fragments, and these DNA fragments are used to reconstruct the restriction site locations of the enzyme. Two common approaches are the Double Digest Problem and the Partial Digest Problem. The Double Digest Problem is known to be NP-Complete[4], but the hardness of the Partial Digest Problem is unknown, despite there being no known polynomial algorithm[5][7].

Alternative approaches to restriction site analysis use *primary fragments*, which are DNA fragments with one endpoint on either the left or right side of the target DNA strand. Blazewicz et. al. [1] [2] present a technique for finding primary fragment lengths called a short digestion, in which the reaction time is chosen so that all molecules are cut at most once. In [6], an alternative approach for finding primary fragment lengths is presented: before the digestion experiment, the endpoints of the molecule are labeled (one method is radio-labeling with radioactive phosphate). [6] uses the primary fragments, in addition to information used in the Partial Digest Problem approach, to find a unique reconstruction in polynomial time. However, the information from the Partial Digest Problem is susceptible to experimental errors caused by missing fragments, and therefore it is still useful to develop techniques based on other types of information that are more robust against these types of errors.

The Simplified Partial Digest Problem, first proposed in [1], uses primary fragments and base fragments to locate restriction sites. Base fragments have two endpoints that were consecutive sites on the target DNA strand and can be obtained by exposing the strand to the enzyme until the digestion process is complete. We consider there are n restriction sites where the enzyme cuts along a DNA strand of length D.

Simplified Partial Digest Problem (SPDP) Statement: Given $X_0 = 0$, $X_{n+1} = D$, and a set of base fragments $\{X_i - X_{i-1}\}_{1 \le i \le n+1}$ and primary fragments $\{(X_{n+1} - X_i) \cup X_i\}_{1 \le i \le n}$, reconstruct the original series $X_1,...,X_n$, where X_i corresponds to the distance between the leftmost end of the target DNA strand and the i^{th} furthest restriction site along the strand.

In this paper, we show that the SPDP is NP-Complete, an open problem in [3]. Therefore, if we desire efficient algorithms for this problem, we must relax our end criteria. We propose an efficient algorithm that in practice finds a solution for many instances of the SPDP.

Results

1) We prove the SPDP is APX-Complete. This is left as an open problem in [3].

We prove this by reducing the Tripartite-Matching problem to it. Given the following Tripartite-Matching instance: *Given sets X,Y,Z with X = Y = Z = {1,2,3..n}, and a set S of triples* $\subseteq X \times Y \times Z$. *Find whether there exists a 'perfect matching' M* $\subseteq S$ *such that |M| = n, and every element of X occurs exactly once in the first term of a triple* $\in M$. *Similarly for Y and Z.* We construct a SPDP as follows:

First, we partition the line into 2T equal-length segments, with T = |S| (these segments are created with symmetric points). There are seven points in each segment on the left half (could be flipped to the other side), three of them are marked with "o" and the other four are marked with "x" (as shown in Figure 1). The i^{th} segment will correspond to the i^{th} triple in the Tripartite-Matching instance. In the i^{th} segment, the "x" points will partition the segment into A - $a_i\varepsilon$, $a_i\varepsilon$, B - $b_i\varepsilon$, $b_i\varepsilon + c_i\varepsilon$, C - $c_i\varepsilon$, the "o" points will partition the segment into four segments I, J, K, and L.

Next, the base fragments are designed such that the four "x" points must be on the same side and the three "o" points must be on the same side by appropriately selecting constants A, B, C, D, I, J, K, and L.

[1]Research supported by NSF/CCR 0113217-001.
[2] Research supported in part by NSF Award 0323766.

Due to space constraints, we leave the description of how to appropriately select these constants to the full version of the paper. We consider the i^{th} triple to be chosen iff the four "x" points and the three "o" points are on different sides of the line. Let the base segments have exactly n copies of I, J, K, L and one copy of A - mε , B - mε , C - mε, m=1, 2, …, n. The lengths of the other 9T-7n additional base fragments can be deduced from the problem instance. We can show that the SPDP we constructed has a solution iff the original Tripartite-Matching problem has a solution.

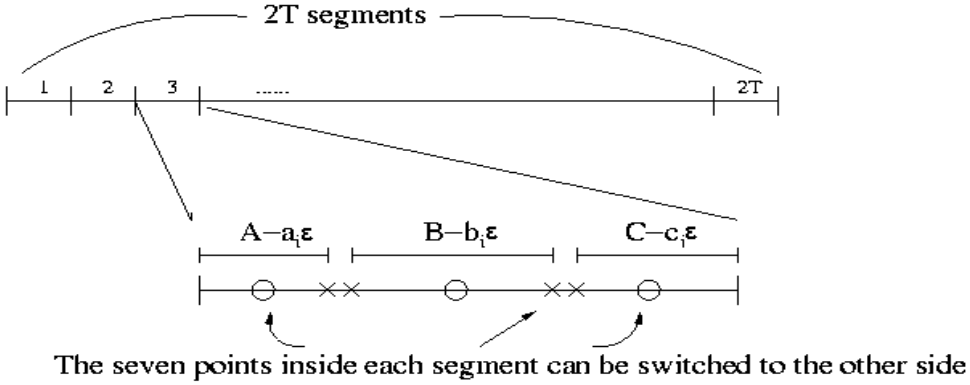

Figure 1

2) We present a new algorithm that solves some instances of the SPDP and has time complexity O(nlogn). We show that for many practical instances of the SPDP, the probability our algorithm finds a solution is above 1/2.

Algorithm:
1. Create an n-element search tree of base fragment lengths.
2. Sort the 2n primary fragments and let the ordered set of the n smallest fragments be S.
3. Set s = 0, current side = left (the opposite of left is right).

For all the elements in S starting with the smallest and increasing in size until S is empty:

3a. We take the next smallest element s' in S and check in the tree whether there exists a base fragment of size s' - s.

3aa. If the base fragment exists, we assign s' to be on the current side and delete one s'-s entry from the search tree.

3ab. If the base fragment does not exist, we assign s' to the opposite side and set the current side to be the opposite side.

3b. Now set s = s' and continue.

One *success condition* for the algorithm is: if there exists a base fragment of size s'-s in step 3aa, then for all other pairs of fragments in S, the absolute value of their difference is *not* equal to s'-s (i.e. no other pair of points in S are distance s'-s apart). If this success condition is true, the algorithm will find a unique solution to the SPDP.

The probability this success condition is met, assuming restriction sites are chosen uniformly at random from the length of the DNA strand, is at least $(1-(1-(1-2/D)^n)^2)^{nD/2}$. For problem instances with $n < 20$, this probability is significant. For instance, with n = 20, D = 20,000, the probability of the success condition is .45. Many practical instances of the SPDP have $n < 20$, since with more than 10 restriction sites, using electrophoresis to distinguish the fragments becomes increasingly difficult [7].

We also note that there is *always* a matching of size at least n/4 between the base fragments generated by the resulting assignment of this algorithm and the base fragments that were given to the algorithm as input. Although this is not useful from the perspective of restriction site analysis, the theoretical insight that our algorithm is a constant approximation to the APX-Complete SPDP suggests that the algorithm probably works well in practice.

Bibliography:

1. J. Blazewicz, P. Formanowicz, M. Kasprzak, M. Jaroszewski, and W.T. Markiewicz. Construction of DNA restriction maps based on a simplified experiment. Bioinformatics Vol. 17 no. 52001, 2001.
2. J. Blazewicz, M. Jaroszewski, and M. Waterman. Combinatorial aspects of the simplified partial digest problem. Recomb, 2003.
3. J. Blazewicz and M. Jaroszewski. New Algorithm for the Simplified partial Digest Problem. WABI, 2003.
4. L. Goldstein and M. Waterman. Mapping DNA by stochastic relaxation. Adv.Appl. Math., 1987.
5. P. Lemke, S. Skiena, and W. Smith. Reconstructing sets from interpoint distances. DIMACS Technical Report, 2002.
6. G.. Pandurangan and H. Ramesh. The restriction mapping problem revisited. Invited paper for a special issue on Computational Biology in Journal of Computer and System Sciences (JCSS), 2001.
7. S. Skiena and G. Sundaram. A partial digest approach to restriction site mapping. ISMB, 1993.