Double Digest Revisited: Complexity and Approximability in the Presence of Noisy Data

Mark Cieliebak[1], Stephan Eidenbenz[2], and Gerhard J. Woeginger[3]

[1] Institute of Theoretical Computer Science, ETH Zurich, `cieliebak@inf.ethz.ch`
[2] Basic and Applied Simulation Science (CCS-5), Los Alamos National Laboratory*, `eidenben@lanl.gov`
[3] Faculty of Mathematical Sciences, University of Twente *and* Department of Mathematics and Computer Science, TU Eindhoven, `g.j.woeginger@math.utwente.nl`

Abstract. We revisit the Double Digest problem, which occurs in sequencing of large DNA strings and consists of reconstructing the relative positions of cut sites from two different enzymes: we first show that Double Digest is strongly NP–complete, improving upon previous results that only showed weak NP–completeness. Even the (experimentally more meaningful) variation in which we disallow coincident cut sites turns out to be strongly NP–complete. In a second part, we model errors in data as they occur in real–life experiments: we propose several optimization variations of Double Digest that model partial cleavage errors. We then show APX–completeness for most of these variations. In a third part, we investigate these variations with the additional restriction that conincident cut sites are disallowed, and we show that it is NP–hard to even find feasible solutions in this case, thus making it impossible to guarantee any approximation ratio at all.

1 Introduction

Double digest experiments are a standard approach to construct physical maps of DNA. Given a large DNA molecule, which for our purposes is an unknown string over the alphabet $\{A, C, G, T\}$, the objective is to find the locations of markers, i.e., occurrences of short substrings such as *GAATTC*, on the DNA. Physical maps are required e.g. in DNA sequencing in order to determine the sequence of nucleotides (A, C, G, and T) of large DNA molecules, since current sequencing methods allow only to sequence DNA fragments with tens of thousands of nucleotides, while a DNA molecule can have up to 10^8 nucleotides.

In double digest experiments, two enzymes are used to cleave the DNA molecule. An enzyme is a protein that cuts a DNA molecule at specific

* LA–UR–03:0532; work done while at ETH Zurich

patterns, the restriction sites. For instance, the enzyme EcoRI cuts at occurrences of the pattern *GAATTC*. Under appropriate experimental conditions, an enzyme cleaves at all restriction sites in the DNA. This process is called *(full) digestion.* Double digest experiments work in three stages: First, clones (copies) of the unknown DNA string are digested by an enzyme A; then a second set of clones is digested by another enzyme B; and finally a third set of clones is digested by a mix of both enzymes A and B, which we will refer to as C. This results in three multisets of DNA fragments. The lengths of these fragments (i.e., their number of nucleotides) are then measured for each multiset by using gel electrophoresis, a standard technique in molecular biology. This leaves us with three multisets of distances (the number of nucleotides) between all adjacent restriction sites, and the objective is to reconstruct the original ordering of the fragments in the DNA molecule, which is the DOUBLE DIGEST problem.

More formally, the DOUBLE DIGEST problem can be defined as follows, where $\mathrm{sum}(S)$ denotes the sum of the elements in a set S, and $\mathrm{dist}(P)$ is the set of all distances between two neighboring points in a set P of points on a line:

Definition 1 (DOUBLE DIGEST)**.** *Given three multisets A, B and C of positive integers with $sum(A) = sum(B) = sum(C)$, are there three sets P^A, P^B and P^C of points on a line, each starting in 0, such that $dist(P^A) = A$, $dist(P^B) = B$ and $dist(P^C) = C$, and such that $P^A \cup P^B = P^C$?*

For example, given multisets $A = \{5, 15, 30\}$, $B = \{2, 12, 12, 24\}$ and $C = \{2, 5, 6, 6, 7, 24\}$ as an instance of DOUBLE DIGEST, then $P^A = \{0, 5, 20, 50\}, P^B = \{12, 14, 26, 50\}$ and $P^C = \{5, 12, 14, 20, 26, 50\}$ is a feasible solution (there may exist more solutions).

Due to its importance in molecular biology, the DOUBLE DIGEST problem has been the subject of intense research since the first successful restriction site mappings in the early 1970's [1, 2]. The DOUBLE DIGEST problem is NP–complete [3], and several approaches including exponential algorithms, heuristics, additional experiments or computer–assisted interactive strategies have been proposed (and implemented) in order to tackle the problem [4–8]. The number of feasible maps for a DOUBLE DIGEST instance can be characterized by using alternating Eulerian paths in appropriate graph classes and can be exponential in the number of fragments [3, 9–11]. For a survey, see [12] and [13].

The double digest experiment is usually carried out with two enzymes that cut at different restriction sites. A majority of all possible enzyme pairings of the more than 3000 known enzymes are pairs with such disjoint cutting behavior. On the other hand, some results in the literature rely on enzymes that cut at the same site in some cases (coincidences) [10]. In particular, NP–hardness of the Double Digest problem has so far only been shown using enzymes that allow for coincidences [3, 12, 14]. Indeed, such enzyme pairs exist, for example enzymes HaeIII and BalI. However, having two enzymes that are guaranteed to always cut at disjoint sites seems more natural and might lead – at least intuitively – to easier reconstruction problems. For example, such instances always fulfill $|C| = |A| + |B| - 1$ (where $|S|$ denotes the cardinality of set S). To reflect these different types of experiments, we define the Disjoint Double Digest problem, which is equivalent to the Double Digest problem with the additional requirement that the two enzymes may never cut at the same site, or, equivalently, that P^A and P^B are disjoint except for the first point (which is 0) and the last point (which is $\text{sum}(A)$).

The NP–hardness results for Double Digest in the literature [3, 12, 14] rely on reductions from weakly NP–complete problems (namely Partition). As a first set of results in this paper, we prove in Section 2 that both Double Digest and Disjoint Double Digest are actually NP–complete in the strong sense by proposing reductions from 3–Partition.

In a second part of the paper, we model reality more closely by taking into account that double digest data usually contains errors. A *partial cleavage* error occurs when an enzyme fails to cut at a restriction site where it is supposed to cut; then one large fragment occurs in the data instead of the two (or even more) smaller fragments. Other error types, such as *fragment length* errors, *missing small fragments*, and *doublets* occur as well (see [5, 7, 6, 14]), but we will focus on partial cleavage errors. They can occur for many reasons, e.g. improper reaction conditions or inaccurate DNA concentration (see e.g. [15] for a list of possible causes). A partial cleavage error occurs e.g. when an enzyme fails to cut at a site where it is supposed to cut in the first (or second) stage of the double digest experiment, but then does cut at this site in the third phase (where it is mixed with the other enzyme). Such an error usually will make it impossible to find a solution for the corresponding Double Digest instance. In fact, only $P^A \cup P^B \subseteq P^C$ can be guaranteed for any solution. Vice–versa, if an enzyme cuts only in the first (or second) phase, but fails to cut in the third phase, then we can only guarantee $P^C \subseteq P^A \cup P^B$.

In the presence of errors, usually the data is such that no exact solutions can be expected. Therefore, optimization criteria are necessary in order to compare and gauge solutions. We will define optimization variations of the Double Digest problem taking into account different optimization criteria; our objective will be to find good approximation algorithms. An optimal solution for a problem instance with no errors will be a solution for the Double Digest problem itself.[4] Thus, the optimization problem cannot be computationally easier than the original Double Digest problem, and (strong) NP–hardness results for Double Digest carry over to the optimization problem.

A straight-forward optimization criterion for Double Digest is to minimize the absolute number of partial cleavage errors in a solution, i.e., to minimize $e(P^A, P^B, P^C) := |(P^A \cup P^B) - P^C| + |P^C - (P^A \cup P^B)|$ (recall that $|S|$ is the cardinality of set S). Here, points in $(P^A \cup P^B) - P^C$ correspond to errors where enzyme A or B failed to cut in the third phase of the experiment, and points in $P^C - (P^A \cup P^B)$ correspond to errors where either enzyme A or B failed to cut in the first resp. second phase. Unfortunately, the corresponding optimization problem Minimum Absolute Error Double Digest in which we try to find point sets P^A, P^B and P^C such that $e(P^A, P^B, P^C)$ is minimum cannot be approximated within any finite approximation ratio (unless $\mathsf{P} = \mathsf{NP}$), as a polynomial-time algorithm guaranteeing a finite approximation ratio could be used to solve the NP–complete Double Digest problem in polynomial-time.

We obtain a more sensible optimization criterion as follows: If we add $|A|+|B|+|C|$ as an offset to the number of errors, we obtain an optimization criterion which turns the absolute number of errors into a measure relative to the input size. The corresponding optimization problem is defined as follows:

Definition 2 (Minimum Relative Error Double Digest). *Given three multisets A, B and C of positive integers with $sum(A) = sum(B) = sum(C)$, find three sets P^A, P^B and P^C of points on a line, each starting in 0, such that $dist(P^A) = A$, $dist(P^B) = B$ and $dist(P^C) = C$, and such that $r(P^A, P^B, P^C) := |A| + |B| + |C| + e(P^A, P^B, P^C)$ is minimum.*

Instead of counting the number of errors, measuring the total size of a solution is an optimization criterion that seems very natural, even if it does not model cleavage errors exactly. In this case, we want to minimize the total number of points in a solution, i.e., we minimize $|P^A \cup P^B \cup P^C|$. This yields the Minimum Point Double Digest problem, which is defined anologous to Minimum Relative Error Double Digest except for the minimization criterion.

[4] Of course, this only holds if the optimization criterion is well–designed.

We show in Section 3 that MINIMUM RELATIVE ERROR DOUBLE DIGEST and MINIMUM POINT DOUBLE DIGEST are APX–hard (i.e., there exists a constant $\varepsilon > 0$ such that no polynomial–time algorithm can guarantee to find approximate solutions that are at most a factor $1+\varepsilon$ off the optimum solution, unless $\mathsf{P} = \mathsf{NP}$) by proposing gap–preserving reductions[5] from MAXIMUM TRIPARTITE MATCHING, using MAXIMUM 4–PARTITION as an intermediary problem. We also analyze a straight–forward approximation algorithm that works for both problems and that achieves an approximation ratio of 2 for MINIMUM RELATIVE ERROR DOUBLE DIGEST and a ratio of 3 for MINIMUM POINT DOUBLE DIGEST.

For each optimization problem, a variation can be defined where the enzymes may only cut at disjoint restriction sites (analogous to DISJOINT DOUBLE DIGEST). The corresponding optimization problems are called MINIMUM DISJOINT RELATIVE ERROR DOUBLE DIGEST and MINIMUM DISJOINT POINT DOUBLE DIGEST. In Section 4, we show that – rather surprisingly – they are even harder to solve than the unrestricted problems: it is NP–hard to even find feasible solutions. We establish this result by showing that the problem of disjointly arranging two given sets of numbers is already NP–hard. This arrangement problem – which we call DISJOINT ORDERING – is a subproblem that every algorithm for any DISJOINT DOUBLE DIGEST variations has to be able to solve; thus, no finite approximation ratio can be achieved for our optimization variations of DISJOINT DOUBLE DIGEST (unless $\mathsf{P} = \mathsf{NP}$). Moreover, the same result would also hold for other optimization criteria, since the proof depends only on the disjointness requirement.

In Section 5, we conclude with directions for future research. Due to space limitations, we only give proof sketches in this extended abstract for most of our results; detailed proofs will be given in the full paper.

2 Strong NP–Completeness of (DISJOINT) DOUBLE DIGEST

In this section, we show strong NP–completeness for the decision problems DOUBLE DIGEST and DISJOINT DOUBLE DIGEST. We present reductions from 3–PARTITION, which is defined as follows: Given $3n$ integers $q_1, \dots, q_{3n}$ and integer h with $\sum_{i=1}^{3n} q_i = nh$ and $\frac{h}{4} < q_i < \frac{h}{2}$ for all $1 \leq i \leq 3n$, are there n disjoint triples of q_i's such that each triple sums up to h? The 3–PARTITION problem is NP–complete in the strong sense [17]. First, we extend the NP–completeness result from [3] for the DOUBLE DIGEST problem.

[5] For an introduction to gap–preserving reductions, see [16].

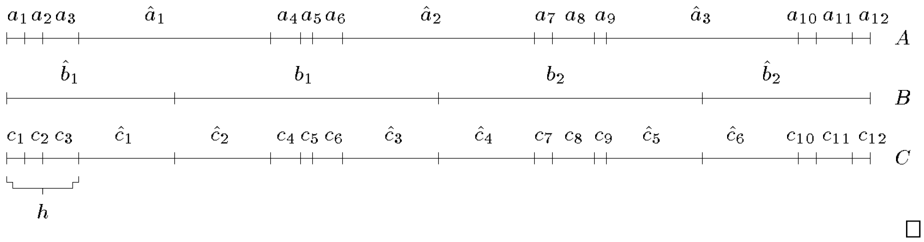
Lemma 3. Double Digest *is strongly* NP*–complete.*

Proof. We reduce 3–Partition to Double Digest as follows: Given an instance $q_1, \ldots, q_{3n}$ and h of 3–Partition, let $a_i = c_i = q_i$ for $1 \leq i \leq 3n$, and let $b_j = h$ for $1 \leq j \leq n$. Then the three (multi-)sets of a_i's, b_j's and c_i's build an instance of Double Digest. If there is a solution for the 3–Partition instance, then there exist n disjoint triples of q_i's (and a_i's as well) such that each triple sums up to h. Starting from 0, we arrange the distances a_i on a line such that each three a_i's that belong to the same triple are adjacent. The same ordering is used for the c_i's. This yields a solution for the Double Digest instance. On the other hand, if there is a solution for the Double Digest instance, say P^A, P^B and P^C, then there exist n subsets of c_i's such that each subset sums up to h, since each point in P^B must occur in P^C as well, and all adjacent points in P^B have distance h. These n subsets yield a solution for the 3–Partition instance. □

Lemma 4. Disjoint Double Digest *is strongly* NP*–complete.*

Proof (sketch). We show strong NP–hardness by reducing 3–Partition to Disjoint Double Digest. Given an instance of 3–Partition, let $s = \sum_{i=1}^{3n} q_i$ and $t = (n+1) \cdot s$. Let $a_i = q_i$ for $1 \leq i \leq 3n$, $\hat{a}_j = 2t$ for $1 \leq j \leq n-1$, $b_j = h + 2t$ for $1 \leq j \leq n-2$, $\hat{b}_k = h + t$ for $1 \leq k \leq 2$, $c_i = q_i$ for $1 \leq i \leq 3n$, and $\hat{c}_j = t$ for $1 \leq j \leq 2n-2$. Let A consist of the a_i's and $\hat{a}_j$'s, and B and C be defined accordingly. Then A, B and C are our instance of Disjoint Double Digest.

Given a solution for the 3–Partition instance, we assume w.l.o.g. that the q_i's (and thus the a_i's and c_i's) are ordered such that the three elements of each triple are adjacent. The arrangement shown in the figure below yields a solution for the Disjoint Double Digest instance. For the opposite direction, let P^A, P^B and P^C be a solution for the Disjoint Double Digest instance. Each two adjacent points in P^B differ by h (plus t or $2t$), and so do $n+1$ points in P^C. Hence, there must be n subsets of c_i's that each sum up to h, yielding a solution for the 3–Partition instance.



□

3 Approximability of Minimum Relative Error Double Digest and Minimum Point Double Digest

In this section, we show that Minimum Relative Error Double Digest and Minimum Point Double Digest are both APX–hard, and we propose a straight–forward approximation algorithm that achieves an approximation ratio of 3 respectively 2 for the two problems. For the proof of APX-hardness, we introduce a maximization variation of the well–known 4–Partition problem [17] which is defined as follows:

Definition 5 (Maximum 4–Partition). *Given an integer h and a multiset $Q = \{q_1, \ldots, q_{4n}\}$ of $4n$ integers with $\sum_{i=1}^{4n} q_i = nh$ and $\frac{h}{5} < q_i < \frac{h}{3}$, find a maximum number of disjoint subsets $S_1, \ldots, S_m \subseteq Q$ such that the elements in each set S_i sum up to h.*

While Maximum 4–Partition may be an interesting problem per se, we are mainly interested in it as an intermediary problem on our way to proving APX–hardness for our optimization variations of Double Digest.

Lemma 6. Maximum 4–Partition *is* APX*–hard.*

Proof (sketch). The lemma follows from the original reduction from Maximum Tripartite Matching to 4–Partition given in [17, pages 97–99], if analyzed as a gap-preserving reduction. □

Lemma 7. Minimum Point Double Digest *is* APX*–hard.*

Proof (sketch). We propose a gap–preserving reduction from Maximum 4–Partition to Minimum Point Double Digest. For a given Maximum 4–Partition instance I, consisting of Q and h, we construct an instance I' of Minimum Relative Error Double Digest as follows: Let $A = C = Q$, and let B contain n times the element h.

Let OPT denote the size of an optimum solution for I, and let OPT' denote the size of an optimum solution for I'. Then we have: if $OPT \geq n$, then $OPT' \leq 4n + 1$, and if $OPT < (1 - \varepsilon)n$ for a small constant $\varepsilon > 0$, then $OPT' > (4 + \frac{\varepsilon}{2})n + 1$. These two implications describe the reduction as gap-preserving and thus establish the result. □

Lemma 8. Minimum Relative Error Double Digest *is* APX*–hard.*

Proof (sketch). The proof uses the same reduction as in Lemma 7 with slightly modified implications. □

A straight–forward approximation algorithm for our two problems simply arranges all distances from A, B and C on a line in a random fashion, starting at 0. If we analyze this algorithm as an approximation algorithm for MINIMUM POINT DOUBLE DIGEST, we see that this will result in a solution with at most $|A| + |B| + |C| - 1$ points; on the other hand, an optimum solution will always use at least $\max(|A|, |B|, |C|) + 1$ points. Thus, this trivial approximation algorithm achieves an approximation ratio of 3 for MINIMUM POINT DOUBLE DIGEST. The same algorithm yields an approximation ratio of 2 for MINIMUM RELATIVE ERROR DOUBLE DIGEST.

4 NP–hardness of Finding Feasible Solutions for Optimization Variations of DISJOINT DOUBLE DIGEST

In this section, we show for all DOUBLE DIGEST optimization variations in which we disallow coincidences that there cannot be a polynomial–time approximation algorithm with finite approximation ratio, unless $\mathsf{P} = \mathsf{NP}$. We achieve this by showing that even finding feasible solutions for these problems is NP–hard. To this end, we introduce the decision problem DISJOINT ORDERING which is defined as follows:

Definition 9 (DISJOINT ORDERING). *Given two multisets A and B of integers with $sum(A) = sum(B)$, find two sets P^A and P^B of points on a line, starting in 0, such that $dist(P^A) = A, dist(P^B) = B$, and such that P^A and P^B are disjoint except for the first and the last point.*

Lemma 10. DISJOINT ORDERING *is* NP*–complete.*

Proof (sketch). Obviously, DISJOINT ORDERING is in NP. To show NP-hardness, we reduce 3–PARTITION to it. Given an instance $q_1, \ldots, q_{3n}$ and h of 3–PARTITION, we construct an instance of DISJOINT ORDERING as follows. Let $a_i = q_i$ for $1 \leq i \leq 3n$, $\hat{a}_j = h$ for $1 \leq j \leq n+1$, $b_i = h+2$ for $1 \leq i \leq n$, and $\hat{b}_j = 1$ for $1 \leq j \leq (n+1) \cdot h - 2n$. Let A consist of the a_i's and $\hat{a}_j$'s, and let B consist of the b_i's and $\hat{b}_j$'s. Then $\mathrm{sum}(A) = \mathrm{sum}(B) = (2n+1) \cdot h$. In the full proof, we show that the following arrangement makes the reduction work: for A, blocks of three a_i's are separated by one $\hat{a}_j$, and for B, each two b_i's are separated by a block of $h-2$ distances $\hat{b}_j$ (with the remaining $\hat{b}_j$'s at the beginning and end). □

We reduce DISJOINT ORDERING to MINIMUM DISJOINT RELATIVE ERROR DOUBLE DIGEST as follows: Let A and B be an instance of DISJOINT ORDERING. We "construct" an instance of MINIMUM DISJOINT

Relative Error Double Digest by simply letting sets A and B be the same sets, and set C be the empty set. If an approximation algorithm for Minimum Disjoint Relative Error Double Digest finds a feasible solution for this instance, this yields immediately a solution for the Disjoint Ordering instance, since any solution feasible solution for Minimum Disjoint Relative Error Double Digest must arrange the elements from A and B in a disjoint fashion. The same argument applies for Minimum Disjoint Point Double Digest, and for any other (reasonable) optimization variation of Disjoint Double Digest since the reduction is totally independent of the optimization criterion. Thus, we have:

Lemma 11. *No polynomial–time approximation algorithm can achieve a finite approximation ratio for* Minimum Disjoint Relative Error Double Digest*,* Minimum Disjoint Point Double Digest*, or any other (reasonable) optimization variation of* Disjoint Double Digest*, unless* $\mathsf{P} = \mathsf{NP}$.

5 Conclusion

In this paper, we showed that Double Digest and Disjoint Double Digest are strongly NP–complete; in a second part, we defined several optimization variations of Double Digest that model partial cleavage errors, proved APX–hardness for Minimum Relative Error Double Digest and Minimum Point Double Digest, and analyzed straight–forward approximation algorithms for these problems that achieve constant approximation ratios. In a last set of results, we showed for Double Digest optimization variations where conincidences are not allowed that even finding feasible solutions is NP–hard.

While our approximability results are tight for all Disjoint Double Digest variations, our results leave considerable gaps regarding the exact approximability threshold for Minimum Relative Error Double Digest and Minimum Point Double Digest, which present challenges for future research. In a different direction of future research, optimization variations of Double Digest that model the three other error types (i.e., fragment length, missing small fragments, and doublets) or even combinations of different error types should be defined and studied. On a meta–level of arguing, it seems unlikely that an optimization variation that models partial cleavage errors *and* some of the other error types could be any easier than the problems that model only partial cleavage errors, but there is a possibility that some error types might offset each other in a cleverly defined optimization problem.

References

1. Smith, H.O., Wilcox, K.W.: A restriction enzyme from hemophilus influenza. I. Purification and general properties. Journal of Molecular Biology **51** (1970) 379–391
2. Danna, K.J., Nathans, D.: Specific cleavage of simian virus 40 DNA by restriction endonuclease of hemophilus influenzal. Proc. of the National Academy of Sciences USA **68** (1971) 2913–2917
3. Goldstein, L., Waterman, M.S.: Mapping DNA by stochastic relaxation. Advances in Applied Mathematics **8** (1987) 194–207
4. Bellon, B.: Construction of restriction maps. Computer Applications in the Biosciences (CABIOS) **4** (1988) 111–115
5. Allison, L., Yee, C.N.: Restriction site mapping is in separation theory. Computer Applications in the Biosciences (CABIOS) **4** (1988) 97–101
6. Wright, L.W., Lichter, J.B., Reinitz, J., Shifman, M.A., Kidd, K.K., Miller, P.L.: Computer–assisted restriction mapping: an integrated approach to handling experimental uncertainty. Computer Applications in the Biosciences (CABIOS) **10** (1994) 435–442
7. Inglehart, J., Nelson, P.C.: On the limitations of automated restriction mapping. Computer Applications in the Biosciences (CABIOS) **10** (1994) 249–261
8. Kao, M.Y., Samet, J., Sung, W.K.: The enhanced double digest problem for DNA physical mapping. In: Proc. of the 7^{th} Scandinavian Workshop on Algorithm Theory (SWAT00). (2000) 383–392
9. Schmitt, W., Waterman, M.S.: Multiple solutions of DNA restriction mapping problems. Advances in Applied Mathematics **12** (1991) 412–427
10. Martin, D.R.: Equivalence classes for the double–digest problem with coincident cut sites. Jounal of Computational Biology **1** (1994) 241–253
11. Pevzner, P.A.: DNA physical mapping and alternating Eulerian cycles in colored graphs. Algorithmica **13** (1995) 77–105
12. Waterman, M.S.: Introduction to Computational Biology. Chapman & Hall (1995)
13. Pevzner, P.A.: Computational Molecular Biology. MIT Press (2000)
14. Setubal, J., Meidanis, J.: Introduction to Computational Molecular Biology. PWS Publishing Company (1997)
15. Promega GmbH `http://www.promega.com/guides/re_guide/toc.htm`: Restriction Enzymes Resource. (2002)
16. Arora, S., Lund, C.: Hardness of approximations. In Hochbaum, D., ed.: Approximation Algorithms for NP–Hard Problems. PWS Publishing Company (1996) 399–446
17. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP–Completeness. Freeman (1979)