Tight bounds for string reconstruction using substring queries

Dekel Tsur*

Abstract

We resolve two open problems presented in [8]. First, we consider the problem of reconstructing an unknown string T over a fixed alphabet using queries of the form "does the string S appear in T?" for some query string S. We show that every non-adaptive algorithm must make $\Omega(\epsilon^{-1/2}n^2)$ queries in order to reconstruct a $1-\epsilon$ fraction of the strings of length n. The second problem is reconstructing a string using queries of the form "does a string from $\mathcal{S}$ appear in T?", where $\mathcal{S}$ is a set of strings. We show a non-adaptive reconstruction algorithm for this model which is optimal both in the number of queries, and in the length of the strings in the queries.

1 Introduction

Consider the following problem: There is some unknown string T of length n over a fixed alphabet, and the goal is to reconstruct T by making queries of the form "does the string S appear in T?" for a *query string* S. Besides the theoretical interest, this problem is motivated by *Sequencing by Hybridization (SBH)* which is a method for sequencing unknown DNA molecules [2]. In this method, the target string is hybridized to a chip containing known strings. For each string in the chip, if its reverse complement appears in the target, then the two strings will bind (or hybridize), and this hybridization can be detected. Thus, SBH can be modeled by the string reconstruction problem described above.

Skiena and Sundaram [11] showed that every string can be reconstructed using $(\sigma-1)n + 2\log n + O(\sigma)$ queries, where σ is the size of the alphabet Σ (assuming that n is known to the algorithm). They also showed a lower bound of $\frac{1}{4}(\sigma-3)n$ queries. The algorithm in [11] is adaptive, namely, each query can depend on the answers to the previous queries. When restricting the algorithm to be non-adaptive (i.e., all of the queries must be determined in advance), the problem becomes harder: At least $\sigma^{n/2}/n$ queries are needed in order to reconstruct all strings of length n, and $O(\sigma^{n/2})$ queries are sufficient [6].

A non-adaptive algorithm for the string reconstruction problem is composed of two parts: The first part is designing a set of queries Q depending on the value of n, and the second part is reconstructing the unknown string given the answers to the queries in Q. In this work, we are mainly interested in the design of the query set. In other words, we are interested whether a given query set provides enough information to solve the reconstruction problem.

The string reconstruction problem becomes easier when relaxing the requirement to reconstruct all strings of length n. For a given query set Q, the *resolution power* $p(Q,n)$ of Q is the fraction of the strings of length n that can be unambiguously reconstructed from the answers to the queries in Q. The query set containing all the strings of length $2\log_\sigma n + \frac{1}{2}\log_\sigma \epsilon^{-1} + O(1)$, called the *uniform query set*, has a resolution power $1-\epsilon$ [1,3,7,10]. The number of queries in this set is $O(\epsilon^{-1/2}n^2)$ (for simplicity, we shall assume that σ is constant).

*Caesarea Rothschild Institute of Computer Science, University of Haifa. Email: dekelts@cs.haifa.ac.il

The question whether the uniform query set is optimal remained open. This problem was posed explicitly in [8, problem 36]. In this paper we settle this open problem by showing that the uniform query set is asymptotically optimal. More precisely, we show that every query set with resolution power at least $1-\epsilon$ contains $\Omega(\epsilon^{-1/2}n^2)$ queries.

Our lower bound also applies to other model which are described below:

Quantitative queries In this model, the queries are of the form “how many times does the string S appear in T?”. Clearly, the $O(\epsilon^{-1/2}n^2)$ upper bound for the uniform query set also applies to this model. Our $\Omega(\epsilon^{-1/2}n^2)$ lower bound is also true in this model.

Set queries In this models, each query is a set of strings, and the answer to the query is whether at least one of the strings in the set appears in T. The *size* of a query set Q is the number of sets (queries) in Q, the *length* of Q is the maximum length of a string that appears in the sets of Q, and the *weight* of a Q is the sum of the sizes of the sets of Q. Frieze et al. [4] showed that every query set Q with resolution power $1-\epsilon$ has size of $\Omega(\log_2(1-\epsilon)+n)$. Moreover, the results of [1,3,7] imply that the length of Q must be at least $2\log_\sigma n+\frac{1}{2}\log_\sigma \epsilon^{-1}$. Our lower bound on the number of queries in the first model above implies that the weight of Q must be $\Omega(\epsilon^{-1/2}n^2)$.

Pevzner and Waterman [8, problem 37] presented the problem of designing a query set which is optimal in its length and in the number of queries. Frieze et al. [4] gave a construction of query sets with (asymptotically) optimal size, but the lengths of these query sets are $\Theta(\log_\sigma^2 n)$. Preparata and Upfal [9] gave an improved analysis for the construction of [4]. From this analysis, it follows that there is a construction of query sets with optimal size, length $3\log_\sigma n+O(1)$, and weight $O(n^3)$ (we note that this result is not explicitly mentioned in [9]). In this paper, we present a new construction of query sets that are optimal (asymptotically) in the size, length, and weight.

Due to lack of space, some proofs are omitted from this abstract. We note that this work focuses on the asymptotic behavior, so the constants in the theorems below were not optimized. We finish this section with some definitions. For a string A, we denote its letters by $a_1, a_2, \ldots$, and we denote by $A[i\colon j]$ the substring $a_i a_{i+1}\cdots a_j$. For two strings A and B, AB is the concatenation of A and B. When we write $\log n$, we assume base σ.

2 The lower bound

In this section, we show the $\Omega(\epsilon^{-1/2}n^2)$ lower bound for the quantitative queries model. This lower bound implies the other lower bounds mentioned in the introduction.

Theorem 1. *There is a constant $\epsilon_0>0$ such that for every $\epsilon\le\epsilon_0$, every query set Q with $p(Q,n)\ge 1-\epsilon$ satisfies $|Q|=\Omega(\epsilon^{-1/2}n^2)$.*

Proof. Consider some $\epsilon\le 1/500$, and let Q be some query set with $p(Q,n)\ge 1-\epsilon$. The main idea of the proof is to partition a subset of the strings of length n into pairs. The paired strings will constitute at least 2ϵ fraction of the strings of length n, and thus Q will reconstruct at least half of these strings. Every paired string that is reconstructed by Q needs to be distinguished from the string it was pair to by some query. By showing that a single query can distinguish between only a small part of the pairs, we will obtain a lower bound on the size of Q. The reason we pair only part of the strings of length n is that some strings have a “complex” structure, and thus handling them is difficult. Pairing only “simple” strings simplifies the analysis.

Let $k=\lfloor 2\log n+\log(1/20\sqrt{\epsilon})\rfloor$. We say that two indices i and j are *far* if $|i-j|\ge 3k+1$, and they are *close* if $|i-j|\le k$. For a string A of length n, a pair of indices (i,j) with $i<j$

is called a *repeat* if $A[i\colon i+k-1] = A[j\colon j+k-1]$. A repeat (i,j) is called *rightmost* if $j \neq n-k+1$ and $(i+1,j+1)$ is not a repeat (i.e., $a_{i+k} \neq a_{j+k}$).

A string u of length k will be called *repetitive* if it has a substring of length $\lceil k/2 \rceil$ that appears at least twice in u. For example, for $k = 6$, the string 'ababac' is repetitive as the string 'aba' appears twice in it. We say that two strings u and v of length k are *similar* if they have a common substring of length $\lceil k/2 \rceil$. An ordered pair (u,v) of dissimilar non-repetitive strings of length k will be called a *simple pair*.

For every simple pair (u,v), let $B_{u,v}$ be the set of all strings A of length n for which there are indices i, i', j, and j' such that

1. $i < i' < j < j'$.
2. Every two indices from $\{i, i', j, j'\}$ are far.
3. (i,j) and (i',j') are rightmost repeats, and there are no other rightmost repeats in A.
4. $A[i\colon i+k-1] = u$ and $A[i'\colon i'+k-1] = v$.

From conditions 3 and 4, we have that the sets $B_{u,v}$ are disjoint. We denote by B the union of the sets $B_{u,v}$ for all simple pairs (u,v).

For a string $A \in B_{u,v}$ whose rightmost repeats are (i,j) and (i',j'), let $\hat{A}$ be the string obtained from A by exchanging the substrings $A[i+k\colon i'-1]$ and $A[j+k\colon j'-1]$, namely,

$$\hat{A} = a_1 \cdots a_{i+k-1} a_{j+k} \cdots a_{j'-1} a_{i'} \cdots a_{j+k-1} a_{i+k} \cdots a_{i'-1} a_{j'} \cdots a_n.$$

We claim that $\hat{A} \in B_{u,v}$. To verify this claim, note that

1. The indices $i_2 = i$, $i'_2 = i + j' - j$, $j_2 = i + j' - i'$, and $j'_2 = j'$ satisfy $i_2 < i'_2 < j_2 < j'_2$.
2. Since every two indices from $\{i, i', j, j'\}$ are far, we obtain that $i'_2 - i_2 = j' - j \geq 3k+1$, $j_2 - i'_2 = j - i' \geq 3k+1$, and $j'_2 - j_2 = i' - i \geq 3k+1$, and therefore every two indices from $\{i_2, i'_2, j_2, j'_2\}$ are far.
3. (i_2, j_2) and (i'_2, j'_2) are rightmost repeats in $\hat{A}$. Moreover, every string of length $k+1$ appears the same number of times in A and in $\hat{A}$. Therefore, if there is a rightmost repeat $(i''_2, j''_2) \neq (i_2, j_2), (i'_2, j'_2)$ in $\hat{A}$, then there are indices i'' and j'' such that $(i'', j'') \neq (i,j), (i',j')$ and (i'', j'') is a rightmost repeat in $\hat{A}$, a contradiction. Thus, (i_2, j_2) and (i'_2, j'_2) are the only rightmost repeats in $\hat{A}$.
4. $\hat{A}[i_2\colon i_2+k-1] = A[i\colon i+k-1] = u$ and $\hat{A}[i'_2\colon i'_2+k-1] = A[i'\colon i'+k-1] = v$.

Lemma 2. *The number of simple pairs is* $(1-o(1))\sigma^{2k}$.

Proof. Let u be a random string of length k. For fixed indices $i < j$, the probability that $u[i\colon i+\lceil k/2 \rceil - 1] = u[j\colon j + \lceil k/2 \rceil - 1]$ is $1/\sigma^{\lceil k/2 \rceil}$. There are $\binom{\lfloor k/2 \rfloor + 1}{2}$ ways to choose the indices i and j. Therefore, the probability that u is repetitive is at most $\binom{\lfloor k/2 \rfloor + 1}{2}/\sigma^{\lceil k/2 \rceil} = O(\log^2 n/n) = o(1)$. Similarly, for two random strings u and v of length k, the probability that u and v are similar is at most $(\lfloor k/2 \rfloor + 1)^2/\sigma^{\lceil k/2 \rceil} = o(1)$. The lemma follows from the two bounds above. ■

Lemma 3. *For every simple pair* (u,v), $|B_{u,v}| \geq \frac{1-o(1)}{48} \left(\frac{\sigma-1}{\sigma}\right)^2 n^4 \sigma^{n-4k}$.

Proof. Fix a simple pair (u, v). Let A be a random string of length n, and let Y be the event that $A \in B_{u,v}$. Our goal is to show that $\mathrm{P}[Y] \geq \frac{1-o(1)}{48}\left(\frac{\sigma-1}{\sigma}\right)^2 n^4/\sigma^{4k}$.

Let I be the set of all pairs $(\alpha = (i,j), \beta = (i',j'))$ such that i, i', j, j' satisfy conditions 1 and 2 in the definition of $B_{u,v}$. For a pair of indices $\alpha = (i,j)$, let Z_α be the event that (i,j) is a rightmost repeat in A, and let Z^w_α be the event that (i,j) is a rightmost repeat and $A[i\colon i+k-1] = w$. For $(\alpha, \beta) \in I$, let $Y_{\alpha,\beta} = Z^u_\alpha \wedge Z^v_\beta \wedge \bigwedge_{\gamma \neq \alpha,\beta} \overline{Z_\gamma}$ (note that the indices of γ can be close). The events $\{Y_{\alpha,\beta}\}_{\alpha,\beta \in I}$ are disjoint, so $\mathrm{P}[Y] = \mathrm{P}\left[\bigvee_{(\alpha,\beta)\in I} Y_{\alpha,\beta}\right] = \sum_{(\alpha,\beta)\in I} \mathrm{P}[Y_{\alpha,\beta}]$. Clearly,

$$\begin{aligned}\mathrm{P}[Y_{\alpha,\beta}] = \mathrm{P}\left[Z^u_\alpha \wedge Z^v_\beta\right] \mathrm{P}\left[\bigwedge_\gamma \overline{Z_\gamma} \middle| Z^u_\alpha \wedge Z^v_\beta\right] &= \mathrm{P}\left[Z^u_\alpha \wedge Z^v_\beta\right]\left(1 - \mathrm{P}\left[\bigvee_\gamma Z_\gamma \middle| Z^u_\alpha \wedge Z^v_\beta\right]\right) \\ &\geq \mathrm{P}\left[Z^u_\alpha \wedge Z^v_\beta\right]\left(1 - \sum_\gamma \mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right]\right).\end{aligned}$$

We now estimate the probabilities in the last expression. As every two indices from α and β are far, we have that the events Z^u_α and Z^v_β are independent, so $\mathrm{P}\left[Z^u_\alpha \wedge Z^v_\beta\right] = \left((\sigma-1)/\sigma^{2k+1}\right)^2$.

To estimate $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right]$, consider some pair of indices $\gamma = (\gamma_1, \gamma_2)$ with $\gamma_1 < \gamma_2$. An index in γ can be close to at most one index in α or β. We consider four cases:

Case 1 If at most one index from γ is close to an index in α or β, then the events Z_γ and $Z^u_\alpha \wedge Z^v_\beta$ are independent, so $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] = \mathrm{P}[Z_\gamma] = (\sigma-1)/\sigma^{k+1}$ (note that $\mathrm{P}[Z_\gamma] = (\sigma-1)/\sigma^{k+1}$ even if the indices of γ are close).

For the rest three cases, we assume that γ_1 is close to an index $j_1 \in \alpha \cup \beta$, and γ_2 is close to an index $j_2 \in \alpha \cup \beta$.

Case 2 Suppose that $j_1 \neq j_2$ and j_1, j_2 are both from α or both from β. If $\gamma_2 - \gamma_1 = j_2 - j_1$ then let $l = \min\{j_1, \gamma_1\}$. The letters $A[l+k]$ and $A[l+j_2-j_1]$ are required to be equal by one of the events Z_γ and $Z^u_\alpha \wedge Z^v_\beta$, and are required to be unequal by the other event. Therefore, $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] = 0$. If $\gamma_2 - \gamma_1 \neq j_2 - j_1$ then we have from [1, p. 437] that the events Z_γ and $Z^u_\alpha \wedge Z^v_\beta$ are independent, so $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] = (\sigma-1)/\sigma^{k+1}$.

Case 3 Suppose that $j_1 = j_2$. W.l.o.g. assume that j_1 is from α. The event Z_γ consists of k letter equality events $A[\gamma_1 + l] = A[\gamma_2 + l]$ for $l = 0, \ldots, k-1$. Let S be the set of all indices l such that the letters $A[\gamma_1 + l]$ and $A[\gamma_2 + l]$ are inside the substring $A[j_1\colon j_1 + k - 1]$. For every $l \leq k-1$, if $l \in S$ then the event $A[\gamma_1 + l] = A[\gamma_2 + l]$ depends on the event $Z^u_\alpha \wedge Z^u_\beta$, and otherwise these events are independent. More precisely, if event Z^u_α happens, then for every $l \in S$ we have that $A[\gamma_1 + l] = A[\gamma_2 + l]$ if and only if $u[\gamma_1 + l - j_1 + 1] = u[\gamma_2 + l - j_1 + 1]$. Thus, $\mathrm{P}[A[\gamma_1 + l] = A[\gamma_2 + l] | Z^u_\alpha]$ is either 0 or 1. If the latter probability is 0 for some l, then $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] = 0$. Otherwise, we have that there is a substring of u of length $|S|$ that appears twice in u. Since u is non-repetitive, we have that $|S| \leq \lceil k/2 \rceil - 1$. The events $A[\gamma_1 + l] = A[\gamma_2 + l]$ for $l \notin S$ are independent, and they are independent of the event $Z^u_\alpha \wedge Z^u_\beta$. Therefore, $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] = 1/\sigma^{k-|S|} \leq 1/\sigma^{\lfloor k/2 \rfloor + 1}$.

Case 4 If one of the indices j_1 and j_2 is from α and the other is from β, we can use similar argument to the one used in case 3, using the fact that u and v are dissimilar. In this case, $\mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] \leq 1/\sigma^{\lfloor k/2 \rfloor+1}$.

The number of pairs γ for which cases 1 or 2 occur is at most $\binom{n}{2}$. The number of pairs γ for which cases 3 or 4 occur is at most $7(2k+1)^2$: There are 4 ways to choose the indices j_1 and j_2 (from $\alpha \cup \beta$) for case 3, and 3 ways to choose these indices in case 4. After choosing j_1 and j_2, there are at most $2k+1$ ways to choose each of the two indices of γ. We conclude that

$$\sum_\gamma \mathrm{P}\left[Z_\gamma | Z^u_\alpha \wedge Z^v_\beta\right] \leq \binom{n}{2} \frac{\sigma-1}{\sigma^{k+1}} + 7(2k+1)^2 \frac{1}{\sigma^{\lfloor k/2 \rfloor+1}} \leq \frac{1}{2}.$$

Since $|I| = \binom{n-3\cdot 3k}{4} = (1-o(1))n^4/24$, it follows that $\mathrm{P}[Y] \geq \frac{1-o(1)}{48}\left(\frac{\sigma-1}{\sigma}\right)^2 n^4/\sigma^{4k}$. ■

We note that the proof of Lemma 3 also implies that $|B_{u,v}| \leq \frac{1}{24}\left(\frac{\sigma-1}{\sigma}\right)^2 n^4 \sigma^{n-4k}$.

From Lemmas 2 and 3 we have that

$$|B| \geq \frac{1-o(1)}{48}\left(\frac{\sigma-1}{\sigma}\right)^2 \frac{n^4}{\sigma^{2k}} \cdot \sigma^n \geq \frac{1-o(1)}{48} \cdot \frac{1}{4} \cdot \frac{n^4}{\sigma^{2k}} \cdot \sigma^n \geq 2\epsilon \cdot \sigma^n.$$

Since $p(Q,n) \geq 1-\epsilon$, it follows that Q reconstructs at least half of the strings in B.

If a string $A \in B$ is uniquely reconstructible by Q, then there must be a query $q \in Q$ that distinguishes between A and $\hat{A}$, that is, q appears a different number of times in A and $\hat{A}$. If q appears more times in A than in $\hat{A}$ we say that q *separates* A. Let M denote the maximum over all $q \in Q$, of the number of strings in B that q separates. From the definition of M we have that the number of strings in Q is at least $\frac{1}{2}|B|/M$. To complete the proof of the theorem, we will give an upper bound on M.

Lemma 4. $M \leq \frac{1}{6}n^4 \cdot \frac{(\sigma-1)^2}{\sigma^{3k+4}} \cdot \sigma^n$.

Proof. If q is a string of length at most $k+1$, then for every string A, q appears the same number of times in A and $\hat{A}$. Thus, q does not separate any string in B. Consider some string $q \in Q$ of length $k+l$ where $l \geq 2$.

In order to bound the number of strings that q separates, consider some random string A. If $A \in B$ and the rightmost repeats of A are (i,j) and (i',j'), then q separates A if and only if one appearance of q in A is a superstring of one of the following substrings of A: $A[i-1 \colon i+k]$, $A[i'-1 \colon i'+k]$, $A[j-1 \colon j+k]$, or $A[j'-1 \colon j'+k]$. In other words, q separates A if and only if $q = A[h \colon h+k+l-1]$ for $h \in \{i-l+1,\ldots,i-1\} \cup \{i'-l+1,\ldots,i'-1\} \cup \{j-l+1,\ldots,j-1\} \cup \{j'-l+1,\ldots,j'-1\}$, and denote the latter event by $X_{(i,j),(i',j')}$.

Consider some fixed h. Recall that Z_α denotes the event that α is a rightmost repeat in A. If $l \leq 2k$, then $\mathrm{P}\left[q = A[h \colon h+k+l-1] \middle| Z_{(i,j)} \wedge Z_{(i',j')}\right] = 1/\sigma^{k+l}$ since the substring $A[h \colon h+k+l-1]$ intersects only one of the substrings $A[i \colon i+k-1]$, $A[i' \colon i'+k-1]$, $A[j \colon j+k-1]$, and $A[j' \colon j'+k-1]$. If $l > 2k$, there we can consider the event that a fixed substring of q of length $3k$ is equal to a corresponding substring of $A[h \colon h+k+l-1]$. The probability of this event, conditioned on the event $Z_{(i,j)} \wedge Z_{(i',j')}$ is $1/\sigma^{3k}$, and therefore, $\mathrm{P}\left[q = A[h \colon h+k+l-1] \middle| Z_{(i,j)} \wedge Z_{(i',j')}\right] \leq 1/\sigma^{3k}$. Thus, for every l, $\mathrm{P}\left[q = A[h \colon h+k+l-1] \middle| Z_{(i,j)} \wedge Z_{(i',j')}\right] \leq 1/\sigma^{k+\min(l,2k)}$. Since there are $4(l-1)$ ways to choose h for fixed i, i', j, and j', it follows that

$$\mathrm{P}\left[X_{(i,j),(i',j')} \middle| Z_{(i,j)} \wedge Z_{(i',j')}\right] \leq \frac{4(l-1)}{\sigma^{k+\min(l,2k)}} \leq \frac{4}{\sigma^{k+2}}.$$

Since the inequality above is true for every i, i', j, and j', we conclude that the number of strings that q separates is at most

$$\frac{4}{\sigma^{k+2}} \cdot \sum_{(\alpha,\beta)\in I} \mathrm{P}\left[Z_\alpha \wedge Z_\beta\right] \cdot \sigma^n \le \frac{4}{\sigma^{k+2}} \cdot \frac{n^4}{24} \cdot \frac{(\sigma-1)^2}{\sigma^{2k+2}} \cdot \sigma^n. \qquad \blacksquare$$

From Lemma 4, the number of strings in Q is at least

$$\frac{|B|}{2M} \ge \frac{\frac{1-o(1)}{48}\left(\frac{\sigma-1}{\sigma}\right)^2 \frac{n^4}{\sigma^{2k}} \cdot \sigma^n}{2 \cdot \frac{1}{6}n^4 \cdot \frac{(\sigma-1)^2}{\sigma^{3k+4}} \cdot \sigma^n} = \Omega(\sigma^k) = \Omega(\epsilon^{-1/2}n^2). \qquad \blacksquare$$

3 Set queries model

We now show a construction of query sets in the set queries model which is optimal in all measures. We will give our result in a slightly different model, called the *gapped queries model*, in which each query is a single word q that contains *gaps*, namely don't care symbols which are denoted by ϕ. The answer to a query q for a string A is "yes" if and only if q matches to a substring of A, where a don't care symbol matches to every symbol. Such a query q can be translated to the set queries model by creating a set q' containing all the words of length $|q|$ that matches to q. For example, if $q = a\phi\phi b$ and the alphabet of A is $\{a, b\}$, then $q' = \{aaab, aabb, abab, abbb\}$.

We will construct a query set Q of the following form: We will choose $k = \log_\sigma n + O(1)$, and build a set $I \subseteq \{1, \dots, 2k\}$ of size k. Then, Q will consists of all the strings q of length $2k$ such that the i-th letter of q is a regular character if $i \in I$ and ϕ if $i \notin I$. Our goal is to choose a set I that maximizes the resolution power of Q.

In order to understand the intuition for building I, consider the following generic reconstruction algorithm that receives as input the answers to the queries of a set Q' on the string A. We assume that for every $q \in Q'$, all the strings in q has length l, and that the first and last $l - 1$ letters of A are known. The algorithm for reconstructing the first $\lceil n/2 \rceil$ letters of A is as follows: (Reconstructing the last $\lfloor n/2 \rfloor$ letters is performed in a similar manner.)

```
1. Let $s_1, s_2, \dots, s_{l-1}$ be the first $l-1$ letters of $A$.
2. For $t = l, l+1, \dots, \lceil n/2 \rceil$ do:
    (a) Let $\mathcal{B}_t$ be the set of all strings $B$ of length $l'$, such that the string $s_1 \cdots s_{t-1}B$ is
        consistent with the answers for $Q'$ on $A$ (i.e., for every $q \in Q'$, if the answer to $q$ on
        $s_1 \cdots s_{t-1}B$ is "yes", then the answer to $q$ on $A$ is "yes").
    (b) If all the strings in $\mathcal{B}_t$ have a common first letter $a$, then set $s_t \leftarrow a$. Otherwise, stop.
3. Return $s_1 \cdots s_{\lceil n/2 \rceil}$.
```

Clearly, for every t, the set $\mathcal{B}_t$ contains the string $a_t \cdots a_{t+l'-1}$. Thus, if the algorithm finishes, then $s_1 \cdots s_{\lceil n/2 \rceil} = a_1 \cdots a_{\lceil n/2 \rceil}$. Moreover, the algorithm stops in iteration t if and only if there is a string $B \in \mathcal{B}_t$ whose first letter is not equal to a_t, and $s_1 \cdots s_{t-1}B$ is consistent with the answers for Q' on A. Such a string B will be called a *bad string (w.r.t. t)*.

Now, suppose that we run the algorithm above with Q' being the uniform query set of size k (that is, all the strings of length k without don't care symbols). Then, the algorithm stops in iteration t if and only if there is a bad string $B' \in \mathcal{B}_t$ such that every substring of $s_1 \cdots s_{t-1}B'$ of length k is also a substring A. The latter events happens if and only if every substring of

$B = s_{t-k+1} \cdots s_{t-1} B'$ of length k is a substring A. Now, the event that the first substring of B of length k is a substring of A has small probability. However, if this event happens, then probability that second substring of B of length k is a substring of A is at least $1/\sigma$: If the first event happens then $B[1\colon k] = A[i\colon i+k-1]$ for some i. Therefore, $A[i+k] = B[k+1]$ is a sufficient condition for the second event, and the probability that $A[i+k] = B[k+1]$ is $1/\sigma$. This "clumping" phenomenon is also true for the other substrings of B, and therefore, the algorithm fails with relatively large probability.

In order to reduce the failure probability, we will use gapped queries as described above, and our goal is to build a set I which will reduce the "clumping" phenomenon. Here, the algorithm fails at iteration t if and only if there is bad a string $B' \in \mathcal{B}_t$ such that every substring of $B = s_{t-2k+1} \cdots s_{t-1} B'$ of length $2k$ is equal to a substring A on the letters of I. The event that the first substring of B is equal to to a substring of A on the letters of I has small probability (note that we need to assume that $2k \in I$ otherwise this event will always occur). The event that the second substring of B is equal to a substring of A on the letters of I still depends on the first event, but now the conditional probability is small: To see this, assume that $B[1\colon 2k]$ is equal to $A[i\colon i+2k-1]$ on the letters of I. Then, the event that $B[2\colon 2k+1]$ is equal to $A[i+1\colon i+2k]$ on the letters of I consists of k letters equalities. Some of this equalities are satisfied due to the fact that $B[1\colon 2k]$ is equal to $A[i\colon i+2k-1]$ on the letters of I, while the other equalities are satisfied with probability $1/\sigma$ each. To have a small probability for the event that $B[2\colon 2k+1]$ is equal to a substring of A on the letters of I, we need that the number of letter equalities of the first kind to be small. This implies the following requirement on I: The intersection of I and $\{x+1 : x \in I\}$ should be small.

We now formalize the idea above. We will define what is a good set I, show that such set exists using the probabilistic method, and then show that if I is a good set, then the set Q has sufficiently large resolution power. We note that Halperin et al. [5] used randomized construction of gapped queries, but their analysis is quite different from ours.

Define $\alpha = \frac{1}{12}$, $\beta = \frac{4}{25}$, $b = 80 + \frac{1}{2}\log_\sigma \frac{8}{\epsilon}$, and $k = \lceil \log_\sigma n + b \rceil$. For an integer x, we denote $I + x = \{y + x : y \in I\}$. We say that a set $I \subseteq \{1, \ldots, 2k\}$ is *good* if it satisfies the following requirements:

1. $|I| = k$.
2. If $X \subseteq \{0, \ldots, \alpha k - 1\}$ and $|X| \geq \frac{5}{8}\alpha k$ then $\bigcup_{x \in X}(I + x) = \{1 + \min(X), \ldots, 2k + \max(X)\}$.
3. For every integer $x \neq 0$, $|I \setminus (I + x)| \geq (\frac{1}{2} - \beta)k$.
4. For every integers $x \neq 0$ and $x' \neq 0$, $|I \setminus ((I + x) \cup (I + x'))| \geq (\frac{1}{4} - \beta)k$.

Lemma 5. *There exists a good set I.*

Proof. Let I be a subset of $\{1, \ldots, 2k\}$ which is built by taking the set $\hat{I} = \{1, \ldots, \alpha k\} \cup \{2k - \alpha k + 1, \ldots, 2k\}$ and then, for each j in $\{\alpha k + 1, \ldots, 2k - \alpha k\}$, j is added to I with probability $(1 - 2\alpha)/(2 - 2\alpha)$. We will show that I is good with positive probability.

Let $m = (2 - 2\alpha)k$ and $l = (1 - 2\alpha)k$. Using Stirling's formula, we obtain that

$$\begin{aligned}
\mathrm{P}\left[|I| = k\right] &= \binom{m}{l}\left(\frac{l}{m}\right)^l\left(1 - \frac{l}{m}\right)^{m-l} = \frac{m!}{l!(m-l)!} \cdot \frac{l^l(m-l)^{m-l}}{m^m} \\
&\geq \frac{\sqrt{2\pi m}\left(\frac{m}{e}\right)^m}{1.09\sqrt{2\pi l}\left(\frac{l}{e}\right)^l \cdot 1.09\sqrt{2\pi(m-l)}\left(\frac{m-l}{e}\right)^{m-l}} \cdot \frac{l^l(m-l)^{m-l}}{m^m} \\
&= \frac{1}{1.09^2\sqrt{2\pi}}\sqrt{\frac{m}{l(m-l)}} \geq \frac{1}{3\sqrt{l}} \geq \frac{1}{3\sqrt{k}},
\end{aligned}$$

namely, the set I satisfies requirement 1 with probability at least $1/(3\sqrt{k})$. We will show that for each other requirement, the probability that it is not satisfied is $2^{-\Omega(k)}$. Therefore, there is a positive probability that all the requirements are satisfied.

Requirement 2 Let j be an index from $\{\alpha k+1,\ldots,2k\}$. Let Y be the number of integers $x \in \{0,\ldots,\alpha k-1\}$ such that $j \in I+x$. For every $x \in \{0,\ldots,\alpha k-1\}$, the probability that $j \in I+x$ is either 1 (if $j \in \hat{I}+x$) or $\frac{1-2\alpha}{2-2\alpha}$. Thus, $\mathrm{E}[Y] \geq \frac{1-2\alpha}{2-2\alpha}\alpha k$. By Chernoff bounds, $Y > \frac{3}{8}\alpha k$ with probability $1-2^{-\Omega(k)}$. Therefore, for a set $X \subseteq \{0,\ldots,\alpha k-1\}$ with size at least $\frac{5}{8}\alpha k$, we have that $j \in \bigcup_{x\in X}(I+x)$. This property holds for every $j \in \{\alpha k+1,\ldots,2k\}$ with probability $1-(2-\alpha)k2^{-\Omega(k)} = 1-2^{-\Omega(k)}$.

Now, for every integers m and M with $0 \leq m < M \leq \alpha k-1$, for every set $X \subseteq \{0,\ldots,\alpha k-1\}$ such that $\min(X) = m$ and $\max(X) = M$, we have that

$$\bigcup_{x\in X}(I+x) \supseteq \bigcup_{x\in X}(\hat{I}+x) = \{1+m,\ldots,\alpha k+M\} \cup \{2k-\alpha k+1+m,\ldots,2k+M\}.$$

Therefore, the probability that requirement 2 is not satisfied is $2^{-\Omega(k)}$.

Requirement 3 Let Y be the number of positions in $I \cap \{\alpha k+1,\ldots,2k-\alpha k\}$ which are not in $I+x$. At least $(2-3\alpha)k$ elements of $\{\alpha k+1,\ldots,2k-\alpha k\}$ are not contained in $\hat{I}+x$. For each such element, the probability that it appears in I and not in $I+x$ is

$$\frac{1-2\alpha}{2-2\alpha}\cdot\frac{1}{2-2\alpha} = \frac{30}{121}.$$

Therefore, $\mathrm{E}[Y] \geq \frac{30}{121}(2-3\alpha)k \geq (\frac{1}{2}-\frac{1}{2}\beta)k$. Using Chernoff bounds, we obtain that the probability that $Y < (\frac{1}{2}-\beta)k$ is $2^{-\Omega(k)}$.

Requirement 4 Using the same arguments as above, we obtain that the probability that $|I \setminus ((I+x)\cup(I+x'))| < (\frac{1}{4}-\beta)k$ is $2^{-\Omega(k)}$. ■

For the rest of the section, we assume that I is good. We use the reconstruction algorithm that is giving in the beginning of this section with $Q' = Q$, $l = 2k$, and $l' = \alpha k$.

Lemma 6. *The probability that the algorithm fails is at most $\epsilon/2$.*

Proof. Fix an iteration t. Recall that a bad string (w.r.t. t) is a string in $\mathcal{B}_t$ whose first letter is not equal to a_t. We will bound the probability that there is a bad string w.r.t. t. We generate a random path $B' = b'_1\cdots b'_{\alpha k}$ as follows: b'_1 is selected uniformly at random from $\Sigma - \{a_t\}$, and for $i > 1$, b'_i is selected uniformly from Σ. Note that each letter of B' has a uniform distribution over Σ.

Let $B = S[t-2k+1\colon t-1]B'$, and denote $B = b_1\cdots b_{2k+\alpha k-1}$. By definition, B' is a bad string if and only if there are indices $r_1,\ldots,r_{\alpha k}$, called *probes*, such that $B[i\colon i+2k-1]$ is equal to $A[r_i\colon r_i+2k-1]$ on the letters of I for all i. We denote by $E(r_i)$ the event corresponding to the probe r_i (i.e. $E(r_i)$ is the event $B[i\colon i+2k-1]$ is equal to $A[r_i\colon r_i+2k-1]$ on the letters of I). Since $b'_1 \neq a_t$, we have that $r_i \neq t-1+i$ for all i.

For a probe r_i, the index i is called the *offset* of the probe, and the value r_i is called the *position* of the probe. A probe r_i is called *close* if $r_i \in \{t-2k+2,\ldots,t\}$. Two probes r_i and $r_{i'}$ will be called *adjacent* if $r_{i'}-r_i = i'-i$, and they will be called *overlapping* if $|r_{i'}-r_i| < 2k$ and they are not adjacent. The adjacency relation is an equivalence relation. We say that an equivalence class of this relation is *close* if it contains a close probe, and we say that two equivalence classes are overlapping if they contain two probes that are overlapping.

Our goal is to estimate the probability that all the events $E(r_i)$ happen. This is easy if these events are independent, but this is not necessarily true. The cause of dependency between the events are close, adjacent, and overlapping probes. We consider several cases (the first six cases are rare cases, while the last two cases are the more common cases):

Case 1 There is an equivalence class which overlaps with at least two different equivalence classes, and all these classes are not close. Let r_{i_1}, r_{i_2} and r_{i_3} be probes from these classes, where r_{i_1} is from the first class, and r_{i_2}, r_{i_3} are from the other two classes. From [1], the events $E(r_{i_1})$ and $E(r_{i_2})$ are independent, so the probability that both events happen is σ^{-2k}. I is good, so it satisfies requirement 4. Event $E(r_{i_3})$ consists of k equalities between a letter of B and a letter of A. From requirement 4, at least $(\frac{1}{4} - \beta)k$ of the letter equalities involve a letter of A that does not participate in the equalities of the events $E(r_{i_1})$ and $E(r_{i_2})$. Therefore, the probability that the events $E(r_i)$, $E(r_{i_1})$, and $E(r_{i_2})$ happen is at least $\sigma^{-(2+\frac{1}{4}-\beta)k}$. The number of ways to choose the positions of the probes r_{i_1}, r_{i_2}, and r_{i_3} is at most $n \cdot (2 \cdot (2+\alpha)k - 1)^2$, and the number of ways to choose the offsets of these probes is $\binom{\alpha k}{3} \leq (\alpha k)^3$. Furthermore, there are at most $n/2$ ways to choose t, and $(\sigma - 1) \cdot \sigma^{\alpha k - 1}$ ways to choose the string B'. Therefore, the overall probability that case 1 happens during the run of the algorithm is at most

$$\frac{n}{2}\sigma^{\alpha k} \cdot \frac{n \cdot ((4+2\alpha)k-1)^2(\alpha k)^3}{\sigma^{(2+\frac{1}{4}-\beta)k}} = O\left(\frac{k^5}{\sigma^{(\frac{1}{4}-\alpha-\beta)k}}\right) = o(1).$$

Case 2 There are two pairs of overlapping equivalence classes, and all these classes are not close. Let r_{i_1}, r_{i_2}, r_{i_3}, and r_{i_4} be probes from these classes, where r_{i_1} and r_{i_2} are from one pair of overlapping classes, and r_{i_3} and r_{i_4} are from the other pair. The events $E(r_{i_1})$ and $E(r_{i_2})$ are independent. Moreover, since r_{i_3} does not overlap with r_{i_1} or r_{i_2} (otherwise, we are in case 1), the event $E(r_{i_3})$ is independent of $E(r_{i_1})$ and $E(r_{i_2})$. Therefore, $\mathrm{P}\left[E(r_{i_1}) \wedge E(r_{i_2}) \wedge E(r_{i_3})\right] = \sigma^{-3k}$. From the fact that I is good, it follows that $\mathrm{P}\left[E(r_{i_4})|E(r_{i_1}) \wedge E(r_{i_2}) \wedge E(r_{i_3})\right] \geq \sigma^{-(\frac{1}{2}-\beta)k}$. Therefore, the probability that case 2 happens is at most

$$\frac{n}{2}\sigma^{\alpha k} \cdot \frac{n^2 \cdot ((4+2\alpha)k-1)^2(\alpha k)^4}{\sigma^{(3+\frac{1}{2}-\beta)k}} = O\left(\frac{k^6}{\sigma^{(\frac{1}{2}-\alpha-\beta)k}}\right) = o(1).$$

Case 3 There are two close equivalence classes. Let r_{i_1} and r_{i_2} be probes from these classes. Then, $\mathrm{P}\left[E(r_{i_1})\right] = \sigma^{-k}$, and $\mathrm{P}\left[E(r_{i_2})|E(r_{i_1})\right] \geq \sigma^{-(\frac{1}{2}-\beta)k}$. Thus, the probability of this case is bounded by

$$\frac{n}{2}\sigma^{\alpha k} \cdot \frac{((2+\alpha)k)^2(\alpha k)^2}{\sigma^{(\frac{3}{2}-\beta)k}} = O\left(\frac{k^4}{\sigma^{(\frac{1}{2}-\alpha-\beta)k}}\right) = o(1).$$

Case 4 There is a pair of overlapping classes, and a close equivalence class. The close class can be either one of the two classes of the overlapping pair, or a third class. In both cases, using the same arguments as above, the probability of such an event is $o(1)$.

If cases 1–4 do not happen, then there is at most one pair of overlapping equivalence classes or one close equivalence class. However, this case does not affect the analysis of the next cases, so we assume in the sequel that there are no overlapping or close probes.

Case 5 The adjacency relation contains 3 equivalence classes each of size at least 2. Let $r_{i_1},\dots,r_{i_6}$ be probes from these classes, where $r_{i_{2j-1}}$ and $r_{i_{2j}}$ are adjacent for $j = 1, 2, 3$. Since these probe do not overlap, we have that the events $E(r_{i_1}) \wedge E(r_{i_2})$, $E(r_{i_3}) \wedge E(r_{i_4})$, and $E(r_{i_5}) \wedge E(r_{i_6})$ are independent. From the fact that I is good it follows that an event $E(r_{i_{2j-1}}) \wedge E(r_{i_{2j}})$ contains at least $(\frac{3}{2}-\beta)k$ distinct letter equalities, so the probability that all the events happen is at least $\left(\sigma^{-(\frac{3}{2}-\beta)k}\right)^3$. Hence, the probability that case 5 happens is

$$\frac{n}{2}\sigma^{\alpha k}\cdot\frac{n^3(\alpha k)^3}{\sigma^{3(\frac{3}{2}-\beta)k}} = O\left(\frac{k^3}{\sigma^{(\frac{1}{2}-\alpha-3\beta)k}}\right) = o(1).$$

Case 6 The adjacency relation contains 2 equivalence classes each of size at least 3. Using similar arguments to the ones in case 5, the probability of this case is at most

$$n\sigma^{\alpha k}\cdot\frac{n^2(\alpha k)^2}{\sigma^{2(\frac{7}{4}-2\beta)k}} = O\left(\frac{k^2}{\sigma^{(\frac{1}{2}-\alpha-2\beta)k}}\right) = o(1).$$

In the following, we assume that cases 5 and 6 do not happen, so there is at most one equivalence class of size at least 3. All the over equivalence classes, except perhaps one class, have size 1.

Case 7 The adjacency equivalence relation contains at least $\frac{\alpha k}{3}$ equivalence classes of size 1. Let $r_{i_1},\dots,r_{i_{\alpha k/3}}$ be the corresponding probes. We have that the events $E(r_{i_j})$ for $j = 1,\dots,\alpha k/3$ are independent, so for fixed probes, the probability of $\bigwedge_{j=1}^{\alpha k/3} E(r_{i_j}) = \sigma^{-k\cdot\alpha k/3}$. For each probe r_{i_j}, there are at most n ways to choose its position. The number of ways to choose the offsets of the probes is $\binom{\alpha k}{\alpha k/3} \le 2^{\alpha k}$. Therefore, the probability of this case is

$$\frac{n}{2}\sigma^{\alpha k}\cdot 2^{\alpha k}\cdot\left(\frac{n}{\sigma^k}\right)^{\frac{\alpha k}{3}} \le n\left(\frac{8}{\sigma^{b-1}}\right)^{\frac{\alpha k}{3}} \le \frac{n}{\sigma^{(b-4)\alpha k/3}} \le \frac{\epsilon}{4}.$$

Case 8 If case 7 does not happen, then there is an equivalence class of size at least $\frac{2}{3}\alpha k - 1 \ge \frac{5}{8}\alpha k$. Let $r_{i_1},\dots,r_{i_s}$ be the probes of this class, and let $r_{j_1},\dots,r_{j_{s'}}$ be the rest of the probes. Denote by m and M the minimal and maximal offsets in $r_{i_1},\dots,r_{i_s}$, respectively, and let $l = m - 1 + \alpha k - M$. By the fact that I is good, we have that the letter equality event of $E(r_{i_1}) \wedge \cdots \wedge E(r_{i_s})$ involves continuous segments of letters in B and A. In other words, $E(r_{i_1}) \wedge \cdots \wedge E(r_{i_s})$ happens if and only if $B[m\colon M+2k-1] = A[p\colon p+2k+M-m-1]$, where p is the minimal position in $r_{i_1},\dots,r_{i_s}$. Therefore, $\mathrm{P}\left[E(r_{i_1}) \wedge \cdots \wedge E(r_{i_s})\right] = \sigma^{-(2k+M-m)} = \sigma^{-((2+\alpha)k-l)}$. Moreover, the event $E(r_{i_1}) \wedge \cdots \wedge E(r_{i_s})$ depends only on the value of m and M and not on the choice of the probes $r_{i_1},\dots,r_{i_s}$.

The events $E(r_{j_1}),\dots,E(r_{j_{s'}})$, perhaps except one, are independent, and each event has probability σ^{-k}. In order to avoid multiplying the probability by the number of ways to choose the offsets of the probes $r_{j_1},\dots,r_{j_{s'}}$, we will consider only the events of probes with offset either less than m or greater than M. The number of such probes is l. For a fixed value of l, there are $l+1$ ways to choose m and M. Therefore, the probability of this case is

$$\sum_{l=0}^{\alpha k}\frac{n}{2}\sigma^{\alpha k}\frac{n}{\sigma^{(2+\alpha)k-l}}\cdot(l+1)\cdot\left(\frac{n}{\sigma^k}\right)^{\max(0,l-1)} \le \frac{1}{\sigma^{2b}}\cdot\sum_{l=0}^{\alpha k}\frac{l+1}{2^l} \le \frac{4}{\sigma^{2b}} \le \frac{\epsilon}{4}. \qquad \blacksquare$$

We obtain the following theorem:

Theorem 7. *For every $\epsilon > 0$ and n, there is a query set Q such that $p(Q,n) \ge 1-\epsilon$, the size of Q is $O(\epsilon^{-1/2}n)$, the length of Q is $2\log_\sigma n + \log_\sigma \frac{1}{\epsilon} + O(1)$, and the weight of Q is $O(\epsilon^{-1}n^2)$.*

In the rest of this section, we consider the time complexity of reconstructing a string A given the answers to the query set defined above. The reconstruction algorithm presented in the beginning of the section can be made more efficient. Instead of checking consistency for every $B \in \mathcal{B}_t$ separately, we do the following: For $i = 1, \dots, \alpha k$, we build the set $\mathcal{B}_t^i$ of all the strings B of length i such that $s_1 \cdots s_{t-1} B$ is consistent with the answers for Q on A. The set $\mathcal{B}_t^i$ (for $i > 1$) is built from $\mathcal{B}_t^{i-1}$ by going over every string $B \in \mathcal{B}_t^{i-1}$ and every character $b \in \Sigma$, and checking whether $s_1 \cdots s_{t-1} Bb$ is consistent. This can be done by considering only one query in Q (the unique query in Q that matches to the suffix of $s_1 \cdots s_{t-1} Bb$ of length $2k$) and checking whether the answer to this query on A is "yes". After building $\mathcal{B}_t^i$, if all the strings in this set has a common first letter, we stop.

Using arguments similar to the ones of 6, we obtain that the expected size of every set $\mathcal{B}_t^i$ is $O(1)$, and moreover, the expected number of sets $\mathcal{B}_t^1, \mathcal{B}_t^2, \dots$ which are built for some t is $O(1)$. It follows that the expected running time of the reconstructing algorithm is $O(nk) = O(n \log_\sigma n)$.

References

[1] R. Arratia, D. Martin, G. Reinert, and M. S. Waterman. Poisson process approximation for sequence repeats, and sequencing by hybridization. *J. of Computational Biology*, 3(3):425–463, 1996.

[2] W. Bains and G. C. Smith. A novel method for nucleic acid sequence determination. *J. Theor. Biology*, 135:303–307, 1988.

[3] M. E. Dyer, A. M. Frieze, and S. Suen. The probability of unique solutions of sequencing by hybridization. *J. of Computational Biology*, 1:105–110, 1994.

[4] A. Frieze, F. Preparata, and E. Upfal. Optimal reconstruction of a sequence from its probes. *J. of Computational Biology*, 6:361–368, 1999.

[5] E. Halperin, S. Halperin, T. Hartman, and R. Shamir. Handling long targets and errors in sequencing by hybridization. In *Proc. 6th Annual International Conference on Computational Molecular Biology (RECOMB '02)*, pages 176–185, 2002.

[6] D. Margaritis and S. Skiena. Reconstructing strings from substrings in rounds. In *Proc. 36th Symposium on Foundation of Computer Science (FOCS 95)*, pages 613–620, 1995.

[7] P. A. Pevzner, Y. P. Lysov, K. R. Khrapko, A. V. Belyavsky, V. L. Florentiev, and A. D. Mirzabekov. Improved chips for sequencing by hybridization. *J. Biomolecular Structure and Dynamics*, 9:399–410, 1991.

[8] P. A. Pevzner and M. S. Waterman. Open combinatorial problems in computational molecular biology. In *Proc. 3rd Israel Symposium on Theory of Computing and Systems, (ISTCS 95)*, pages 158–173, 1995.

[9] F. Preparata and E. Upfal. Sequencing by hybridization at the information theory bound: an optimal algorithm. *J. of Computational Biology*, 7:621–630, 2000.

[10] R. Shamir and D. Tsur. Large scale sequencing by hybridization. *J. of Computational Biology*, 9(2):413–428, 2002.

[11] S. Skiena and G. Sundaram. Reconstructing strings from substrings. *J. of Computational Biology*, 2:333–353, 1995.