Finding all Palindrome Subsequences on a String

K.R. Chuang[1], R.C.T. Lee[2] and C.H. Huang[3*]
[1, 2] Department of Computer Science, National Chi-Nan University, Puli, Nantou Hsieh, Taiwan 545
[3] Department of Computer Science and Information Engineering, National Formosa University, 64, Wen-Hwa Road, Hu-wei, Yun-Lin, Taiwan 632
[*]Corresponding author: chhuang@sunws.nfu.edu.tw

Abstract

Palindromes are strings of symbols that read the same forward and backward. In DNA sequences, palindromes appear frequently and are widespread in human cancers and identifying them could help advance the understanding of genomic instability. The palindrome detection problem is therefore an important issue in computational biology. In this paper, we propose the finding all palindrome subsequences problem and give an algorithm to find all palindrome subsequences.

Keywords: Palindrome, Palindrome Subsequence

Section 1 Introduction

In this paper, the following notations are used. A string is a sequence of symbols from an alphabet set. For a string $S = s_1s_2\ldots s_n$ of length n, let s_i denote the ith symbol in S. A subsequence of S is obtained by deleting zero or more (not necessarily consecutive) symbols from S.

Palindromes are strings of the form ww^R or waw^R where w is a non-empty substring, w^R is the reverse of w^R and a is non-empty symbol. If we have strings in the form ww^R, we call these strings even palindromes. If we have strings in the form waw^R, we call these things odd palindromes. For example, GG and TAGGAT are both even palindromes. ATA and GATCTAG are odd palindromes.

In computational molecular biology, exploring DNA function is very important. When exploring DNA function, special subsequences such as palindromes may represent important messages. In DNA sequences, palindromes appear frequently and are widespread in human cancers. Identifying palindromes of DNA sequences could help advance the understanding of genomic instability [5, 6]. The finding palindromes problem is therefore an important issue in computational biology. There have been many researches on the palindromes finding problem and there are many various classic computing problems on the palindromes finding problem. Manacher discovered an on-line sequential algorithm that finds all initial palindromes in a string [3]. Dan Gusfield gave a linear-time algorithm to solve the finding all maximal palindromes problem in a string [2]. Porto and Barbosa gave an algorithm to find all approximate palindromes in a string [1].

The palindromes which the above algorithms found are substrings of a given string. In this paper, we pay attention to the palindrome subsequence. The palindrome subsequence is defined as following. Given a string $S = s_1s_2\ldots s_n$, a palindrome subsequence of S is a subsequences of S which is a palindrome. For example, we suppose that S = ACGATGTAC. AGGA is a palindrome

subsequence of *S*. We propose the finding all palindrome subsequences problem and give an algorithm to solve it.

Section 2 **The Method**

To begin with, we introduce the property of palindrome. Let $P = p_1p_2\ldots p_{m-1}\ p_m$ be a palindrome and (p_i, p_j) be a matched pair where p_i and p_j are identical character and $1 \le i < j \le m$. If P is an even palindrome of length m, P consists of $\frac{m}{2}$ matched pairs such as (p_1, p_m) (p_2, p_{m-1}) … $(p_{\frac{m}{2}}, p_{\frac{m}{2}-1})$. For example, P = ATTA is an even palindrome which consists of 2 matched pairs such as (p_1, p_4) (p_2, p_3), shown in Figure 2-1. If P is an odd palindrome of length m, P consists of $\left\lfloor \frac{m}{2} \right\rfloor$ matched pairs and a central symbol p_c. For example, P = ATGTA is an odd palindrome which consists 2 matched pairs such as (p_1, p_5) (p_2, p_4) and p_3 is the central symbol, shown in Figure 2-2.

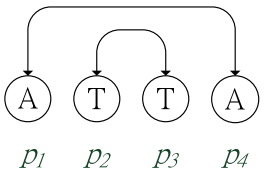
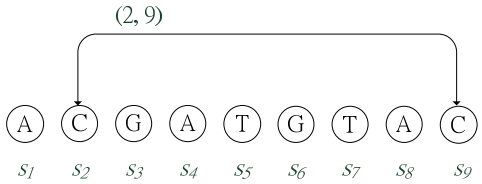


Figure 2-1

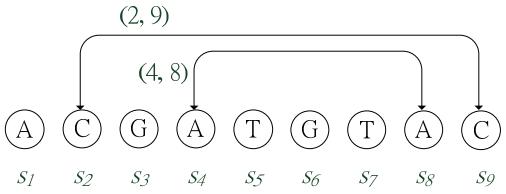
Palindrome subsequences also have the same property with palindromes, because palindrome subsequences are palindromes. Given a string $S = s_1s_2\ldots s_{n-1}\ s_n$ of length n, let (i, j) be the match pair where i and j denote that s_i is matched with s_j and $1 \le i < j \le n$. Let (i_1, j_1)-(i_2, j_2)- … -(i_k, j_k) denote even palindrome subsequences of S with k matched pairs where $1 \le i_1 < i_2 < \ldots i_k < j_k < \ldots < j_2 < j_1 \le n$ and $k < \frac{n}{2}$. If an even palindrome subsequence consists k matched pair, we call it a k-pair palindrome subsequence. For example, given a string S = CGATGTAC, ATTA, "(3, 7)-(4, 6)", is a 2-pair palindrome subsequence of S. Let (i_1, j_1)-(i_2, j_2)- … -(i_k, j_k)-c denote odd palindrome subsequences of S with k matched pairs and one central symbol s_c where c is the position of s_c on S and $i_k < c < j_k$. The odd palindrome subsequences with k matched pairs can be obtained from k-pair palindrome subsequences with one central symbol. Odd palindrome subsequences could be found easily, when all even palindrome subsequences are found. For example, given a strng S = CGATGTAC, ATGTA, "(3, 7)-(4, 6)-5", composes of 2-pair palindrome subsequence, "(3, 7)-(4, 6)" and one central symbol s_5. There may be too many odd palindrome subsequences based on an even palindrome subsequences, so we only find the even palindrome subsequences in this paper.

The k-pair palindrome subsequence has a property. The k-pair palindrome subsequence composes of a k-1-pair palindrome subsequence and 1-pair palindrome subsequence. Let k-1-palindrome be (i_1, j_1)- … -(i_{k-1}, j_{k-1}) and 1-pair palindrome subsequence be (i', j'). The k-pair palindrome subsequence, (i_1, j_1)- … -(i_{k-1}, j_{k-1})-(i', j'), can be found from k-1-pair palindrome subsequence and 1-pair palindrome subsequence where $i' > i_{k-1}$ and $j' < j_{k-1}$. For example, given a string S = ACGATGTAC, CC, CAAC and CATTAC are palindrome subsequences of S. CC is a 1-pair palindrome subsequence, "(2, 9)", shown in Figure 2-2 (a), AA is also a 1-pair palindrome subsequence, "(4, 8)" and TT is also a 1-pair palindrome subsequence, "(5, 7)". CAAC is a 2-pair palindrome subsequence, "(2, 9)-(4, 8)" which composes of two 1-pair palindrome

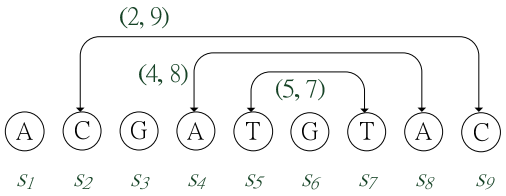
subsequences (2, 9) and (4, 8), shown in Figure 2-2 (b). CATTAC is a 3-pair palindrome subsequence, “(2, 9)-(4, 8)-(5, 7)”, which composes of a 2-pair palindrome subsequence, “(2, 9)-(4, 8)”, and a 1-pair palindrome subsequence, “(5, 7)”, shown in Figure 2-2 (c).



(a) The matched pair of CC



(b) The matched pairs of CAAC



(c) The matched pairs of CATTAC

Figure 2-2

According to the above property of *k*-pair palindrome subsequence, we can use it to find all palindrome subsequences. For example, given a string *S* = ACGATGTAC, we can use it to find all palindrome subsequences of *S* as follows:

S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9

A C G A T G T A C

First, we find all matched pairs of *S* and each matched pair is a 1-pair palindrome subsequence.

(1, 4) AA

(1, 8) AA

(2, 9) CC

(3, 6) GG

(4, 8) AA

(5, 7) TT

After all 1-palindrome subsequences of *S* are found, we can find all 2-palindrome subsequences based upon them.

(1, 8)-(3, 6) AGGA

(1, 8)-(5, 7) ATTA

(2, 9)-(3, 6) CGGC

(2, 9)-(4, 8) CAAC

(2, 9)-(5, 7) CTTC

(4, 8)-(5, 7) ATTA

After finding all 2-palindrome subsequences, we can find all 3-palindrome subsequences based upon 2-palindrome subsequence and 1-palindrome subsequence.

(2, 9)-(4, 8)-(5, 7) CATTAC

The recursive process continues until all palindrome subsequence are found out.

Section 3 The Algorithm

We proposed an algorithm to solve the finding all palindrome subsequences problem. In this algorithm, we find all palindrome subsequences from one palindrome subsequence to the longest palindrome subsequence. Given a string S of length n, let U_k be the set of k-pair palindrome where $1 \le k \le \frac{n}{2}$.

Step 1: We use incidence matrix to find all matched pairs (i, j) where $1 \le i < j \le n$ and add them into U_1, because each matched pair is 1-pair palindrome subsequence.

Step 2: We generate U_k from U_{k-1} and U_1 where $1 \le k \le \frac{n}{2}$. For all k-1-pair palindrome subsequences in U_{k-1}, we take a k-1-pair palindrome subsequence (i_1, j_1)- … -(i_{k-1}, j_{k-1}) from U_{k-1} and we check all 1-pair palindromes from U_1 whether there is a 1-pair palindrome (i', j') which satisfies the rule $i' > i_{k-1}$ and $j' < j_{k-1}$. If it is satisfied, we combine the k-1-pair palindrome (i_1, j_1)- … -(i_{k-1}, j_{k-1}) with the 1-pair palindrome (i', j') to be k-pair palindrome (i_1, j_1)- … -(i_{k-1}, j_{k-1})-(i', j') and add it into the set U_k. Until the $U_{n/2}$ is generated, we can get the set $U = U_1 \cup U_2 \cup \ldots \cup U_{n/2}$ which contains all palindrome subsequences of S.

In the following, we present the algorithm for finding all palindrome subsequences.

Algorithm *findAllPalindromeSubsequences*(S)

Input: A string $S = s_1 s_2 \ldots s_n$.

Output: All palindrome subsequences of S.

Step 1:

/* Finding out matched pair for $1 \le i < j \le n$ */

$U_1 := \{\}$

for $i = 1$ **to** n **do**

  for $j = i + 1$ **to** n **do**

    if $s_i = s_j$ **then**

      $w := (i, j)$

      $U_1 := U_1 \cup \{w\}$

  endfor

endfor

Step 2:

/* Finding all palindrome subsequences of S */

for $k = 2$ **to** $n/2$ **do**

  $U_k := \{\}$

  for all k-1-pair palindrome $(i_1, j_1) \ldots (i_{k-1}, j_{k-1})$ from U_{k-1} **do**

    for all 1-pair palindrome (i', j') from U_1 **do**

      if $i' > i_{k-1}$ and $j' < j_{k-1}$ **then**

        $i_k := i'$

        $j_k := j'$

        $w := (i_1, j_1) \ldots (i_{k-1}, j_{k-1}) (i_k, j_k)$

        $U_k := U_k \cup \{w\}$

endif

endfor

endfor

endfor

$U := U_1 \bigcup U_2 \bigcup \ldots \bigcup U_{n/2}$

/* U is the set of all palindrome subsequences of S */

Obviously, the time complexity of this sample algorithm is $O(n^3)$ where n is the length of the sequence.

Section 4 An Example

Given a string S = ACGATGTAC, We now illustrate the whole procedure in detail.

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
A	C	G	A	T	G	T	A	C

Step 1: We use incidence matrix to find all matched pairs (i, j) where $1 \le i < j \le n$.

Table 1 The incidence matrix for this sequence S = ACGATGTAC

S_i \ S_j		1	2	3	4	5	6	7	8	9
		A	C	G	A	T	G	T	A	C
1	A		0	0	1	0	0	0	1	0
2	C			0	0	0	0	0	0	1
3	G				0	0	1	0	0	0
4	A					0	0	0	1	0
5	T						0	1	0	0
6	G							0	0	0
7	T								0	0
8	A									0
9	C									

After the incidence matrix is generated, we can get the U_1.

U_1 = {(1, 4), (1, 8), (2, 9), (3, 6), (4, 8), (5, 7)}

Step 2:

(1) k = 2, U_1 = {(1, 4), (1, 8), (2, 9), (3, 6), (4, 8), (5, 7)}, U_2 = {}

(1-1)

We take the 1-palindrome subsequence (1, 4) from U_1.

For all 1-pair palindrome subsequences from U_1, there is no 1-pair palindrome subsequence (i', j') which satisfies that $i' > 1$ and $j' < 4$.

U_2 = {}

(1-2)

We take the 1-palindrome subsequence (1, 8) from U_1.

For all 1-pair palindrome subsequences from U_1, there is a 1-pair palindrome subsequence (3, 6) which satisfies that 3 > 1 and 6 < 8. We combine (1, 8) with (3, 6) to be 2-pair palindrome subsequence (1, 8)-(3, 6) and add it into the set U_2.

U_2 = {(1, 8)-(3, 6)}

There is another 1-pair palindrome subsequence (5, 7) which can satisfy that 5 > 1 and 7 < 8. We

combine (1, 8) with (5, 7) to be 2-pair palindrome subsequence (1, 8)-(5, 7) and add it into the set U_2.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7)}

There is no 1-pair pair palindrome subsequence which can be satisfied.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7)}

(1-3)

We take the 1-pair palindrome subsequence (2, 9) from U_1.

There is a 1-pair palindrome subsequence (3, 6) which can be satisfied. We combine (2, 9) with (3, 6) to be 2-pair palindrome subsequence (2, 9)-(3, 6) and add it into the set U_2.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6)}

There is another 1-pair palindrome subsequence (4, 8) which can be satisfied. We combine (2, 9) with (4, 8) to be 2-pair palindrome subsequence (2, 9)-(4, 8) and add it into the set U_2.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6), (2, 9)-(4, 8)}

There is another 1-pair palindrome subsequence (5, 7) which can be satisfied. We combine (2, 9) with (5, 7) to be 2-pair palindrome subsequence (2, 9)-(5, 7) and add it into the set U_2.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6), (2, 9)-(4, 8), (2, 9)-(5, 7)}

There is no 1-pair palindrome subsequence which can be satisfied.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6), (2, 9)-(4, 8), (2, 9)-(5, 7)}

(1-4)

We take the 1-pair palindrome subsequence (3, 6) from U_1.

Check all 1-pair palindromes from U_1.

There is no 1-pair palindrome which can be satisfied.

U_2 = {(1, 8) (3, 6), (1, 8) (5, 7), (2, 9) (3, 6), (2, 9) (4, 8), (2, 9) (5, 7)}

(1-5)

We take the 1-pair palindrome (4, 8) from U_1.

Check all 1-pair palindromes from U_1.

There is a 1-pair palindrome (5, 7) which can be satisfied. We combine (4, 8) with (5, 7) to be 2-pair palindrome (4, 8)-(5, 7) and add it into the set U_2.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6), (2, 9)-(4, 8), (2, 9)-(5, 7), (4, 8)-(5, 7)}

There is no 1-pair palindrome which can be satisfied.

U_2 = {(1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6), (2, 9)-(4, 8), (2, 9)-(5, 7), (4, 8)-(5, 7)}

(1-6)

We take the 1-pair palindrome (5, 7) from U_1.

Check all 1-pair palindromes from U_1.

There is no 1-pair palindrome which can be satisfied.

(2) $k = 3$, $U_1 = \{(1, 4), (1, 8), (2, 9), (3, 6), (4, 8), (5, 7)\}$, $U_2 = \{(1, 8)\text{-}(3, 6), (1, 8)\text{-}(5, 7), (2, 9)\text{-}(3, 6), (2, 9)\text{-}(4, 8), (2, 9)\text{-}(5, 7), (4, 8)\text{-}(5, 7)\}$, $U_3 = \{\}$

(2-1)

We take the 2-pair palindrome (1, 8)-(3, 6) from U_2.

Check all 1-pair palindrome from U_1.

There is no 1-pair palindrome which can be satisfied.

$U_3 = \{\}$

(2-2)

We take the 2-pair palindrome (1, 8)-(5, 7) from U_2.

Check all 1-pair palindrome from U_1.

There is no 1-pair palindrome which can be satisfied.

$U_3 = \{\}$

(2-3)

We take the 2-pair palindrome (2, 9)-(3, 6) from U_2.

Check all 1-pair palindrome from U_1.

There is no 1-pair palindrome which can be satisfied.

$U_3 = \{\}$

(2-4)

We take the 2-pair palindrome (2, 9)-(4, 8) from U_2.

Check all 1-pair palindrome from U_1.

There is a 1-pair palindrome (5, 7) which can be satisfied. We combine (2, 9)-(4, 8) with (5, 7) to be 3-pair palindrome (2, 9)-(4, 8)-(5, 7) and add it into the set U_3.

$U_3 = \{(2, 9)\text{-}(4, 8)\text{-}(5, 7)\}$

(2-5)

We take the 2-pair palindrome (2, 9)-(5, 7) from U_2.

Check all 1-pair palindrome from U_1.

There is no 1-pair palindrome which can be satisfied.

$U_3 = \{(2, 9)\text{-}(4, 8)\text{-}(5, 7)\}$

(2-6)

We take the 2-pair palindrome (4, 8)-(5, 7) from U_2.

Check all 1-pair palindrome from U_1.

There is no 1-pair palindrome which can be satisfied.

(3) $k = 4$, $U_1 = \{(1, 4), (1, 8), (2, 9), (3, 6), (4, 8), (5, 7)\}$, $U_2 = \{(1, 8)\text{-}(3, 6), (1, 8)\text{-}(5, 7), (2, 9)\text{-}(3, 6), (2, 9)\text{-}(4, 8), (2, 9)\text{-}(5, 7), (4, 8)\text{-}(5, 7)\}$, $U_3 = \{(2, 9)\text{-}(4, 8)(5, 7)\}$, $U_4 = \{\}$

(3-1)

We take the 3-palindrome (2, 9)-(4, 8)-(5, 7) from U_3.

Check all 1-pair palindrome from U_1.

There is no 1-pair palindrome which can be satisfied.

$U_4 = \{\}$

Finally, we get the set $U = U_1 \bigcup U_2 \bigcup \ldots \bigcup U_{n/2}$ which contains all palindrome subsequences of S.

U = {(1, 4), (1, 8), (2, 9), (3, 6), (4, 8), (5, 7), (1, 8)-(3, 6), (1, 8)-(5, 7), (2, 9)-(3, 6), (2, 9)-(4, 8), (2, 9)-(5, 7), (4, 8)-(5, 7), (2, 9)-(4, 8)-(5, 7)}

The all palindrome subsequences of S are as follows:

(1, 4) AA

(1, 8) AA

(2, 9) CC

(3, 6) GG

(4, 8) AA

(5, 7) TT

(1, 8)-(3, 6) AGGA

(1, 8)-(5, 7) ACCA

(2, 9)-(3, 6) CGGC

(2, 9)-(4, 8) CAAC

(2, 9) (5, 7) CTTC

(4, 8)-(5, 7) ATTA

(2, 9)-(4, 8)-(5, 7) CATTAC

Section 5 Conclusions and Future Work

In this paper, we proposed an algorithm to solve the finding all palindrome subsequences in a string. Palindrome subsequences occur frequently in DNA sequences and have been proved to be critical for some characteristics. Our algorithm provides an effective tool for the related research.

References

[1] Alexandre H.L. Porto, Valmir C. Barbosa, Finding Approximate Palindromes in Strings. Pattern Recognition, 2002.

[2] D. Gusfied, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, New York, 1997.

[3] G. Manacher. A new Linear-Time “On-Line” Algorithm for Finding the Smallest Initial Palindrome of a String. J. Assoc. Comput. 1975

[4] Lloyd Allison, Finding Approximate Palindromes in Strings Quickly and Simply

[5] Choi, Charles Q, DNA palindromes found in cancer, The Scientist 2005.

[6] Tanaka, Hisashi; BERGSTROM, Donald A; YAO, Meng-Chao and TAPSCOTT, Stephen J, Large DNA palindromes as a common form of structural chromosome aberrations in human cancers, Human Cell, 2006