Finding all Palindrome Subsequences in a String

K.R. Chuang1, R.C.T. Lee2 and C.H. Huang3*
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 Rod, Hu-wei, Yun-Lin, Taiwan 632
*Corresponding author: chhuang@sunws.nfu.edu.tw

Abstract

A palindrome is a string of symbols that is read the same forward and backward. Palindrome also occurs in DNA. DNA palindromes appear frequently and are widespread in human cancers. Identifying them could help advance the understanding of genomic instability [2, 6]. The Palindrome subsequences detection problem is therefore an important issue in computational biology. In this paper, we present an algorithm to find all palindrome subsequences.

1. Introduction

In this paper, the following notations are used. A string is a sequence of symbols from an alphabet set $\sum$. For a string $S = s_1s_2...s_n$ 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 form S.

A palindrome is a string of the form ww^R where w is a non-empty substring and w^R is the reverse of w. For example, TT and GCAACG are palindromes. There are many various classic computing problems in finding palindromes of a string. For example, Manacher discovered an on-line sequential algorithm that finds all initial palindromes in a string [4]. Porto and Barbosa gave an algorithm to find long approximate palindromes [5].

Given a string S, a subsequence P is a palindrome subsequence of S if P is a palindrome. Taking a string S = ACGATGTAC as an example, a palindrome subsequence of S is ATTA. In computational molecular biology, finding out the palindrome subsequences in DNA sequence is an important issue [3]. However, as far as we know, there is no article discussing about how to detect all palindrome subsequences. In this paper, we proposed an effective algorithm to solve the palindrome subsequence problem.

2. The Method

To begin with, we introduce an idea from the properties of palindrome. Let $P = p_1p_2...p_m$ be a palindrome. If P is a palindrome, p_1 is matched with p_m and p_2 is matched with p_{m-1} and so forth. For example, given a palindrome P = ATTA, p_1 is matched with p_4 and p_2 is matched p_3 (Figure 1). Palindrome subsequences also have the same property of palindrome, because palindrome subsequences are palindromes.

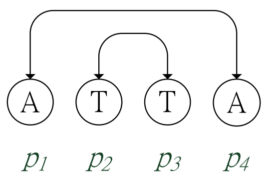


Figure 1

Let matched pair, (i, j), to denote that s_i is matched with s_j where $1 \le i < j \le n$ and we define k-palindrome subsequence to be a palindrome subsequence which has k matched pairs of S. We use the notation (i_1, j_1) (i_2, j_2) … (i_k, j_k) to denote k-palindrome subsequence where $1 \le i_1 < i_2 < ...i_k < j_k < ... < j_2 < j_1 \le n$. Given a string S = ACGATGTAC, AGGA is one of all palindrome subsequences of S. The matched pairs of AGGA are (1, 8) and (3, 6) (Figure 2). It is a 2-palindrome subsequence which is denoted as (1, 8) (3, 6).

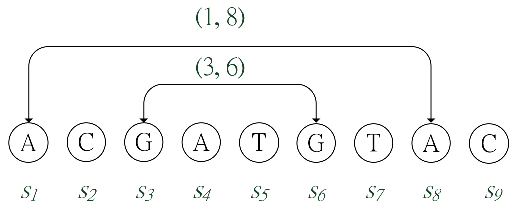
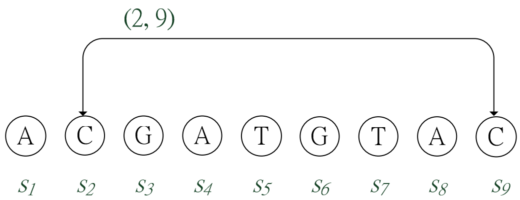
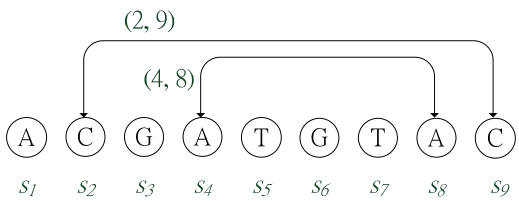


Figure 2

The k-palindrome subsequence has one property which is that the k-palindrome subsequence is based up on k-1-palindrome subsequence and 1-palindrome subsequence. Let k-1-palindrome be $(i_1, j_1) \ldots (i_{k-1}, j_{k-1})$ and 1-palindrome subsequence be (i', j'). The k-palindrome subsequence, $(i_1, j_1) \ldots (i_{k-1}, j_{k-1}) (i', j')$, can be found from k-1-palindrome subsequence and 1-palindrome subsequence, if the $i' > i_{k-1}$ and $j' < j_{k-1}$. For example, given a string S = ACGATGTAC then CC, CAAC and CATTAC are palindrome subsequences of S. CC is a 1-palindrome subsequence denoted (2, 9) (Figure 3(a)), AA is also a 1-palindrome subsequence denoted (4, 8) and TT is also a 1-pailindrome subsequence denoted (5, 7). CAAC is a 2-palindrome subsequence denoted (2, 9) (4, 8) which is based upon 1-palindrome subsequence (Figure 3(b)). CATTAC is a 3-palindrome subsequence denoted (2, 9) (4, 8) (5, 7) which is based upon 2-palindrome subsequence and 1-palindrome subsequence.

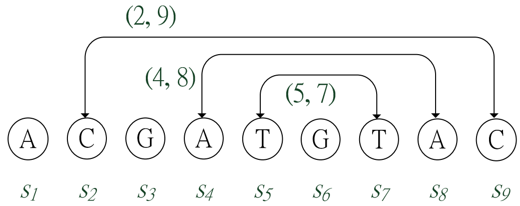


(a) The matched pair of CC



(b) The matched pairs of

CAAC



(c) The matched pairs of CATTAC

Figure 3

According to the above property of k-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-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.

3. The Algorithm

We proposed an algorithm to solve the

finding all palindrome subsequences problem. In this algorithm, we find all palindrome subsequences form one palindrome subsequence to the longest palindrome subsequence. Given a string S of length n, let U_k be the set of k-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-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-palindrome subsequences in U_{k-1}, we take a k-1-palindrome subsequence $(i_1, j_1) \ldots (i_{k-1}, j_{k-1})$ form U_{k-1} and we check all 1-palindromes from U_1 whether there is a 1-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-palindrome $(i_1, j_1) \ldots (i_{k-1}, j_{k-1})$ with the 1-palindrome (i', j') to be k-palindrome $(i_1, j_1) \ldots (i_{k-1}, j_{k-1}) (i', j')$ 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-palindrome $(i_1, j_1) \ldots (i_{k-1}, j_{k-1})$ from U_{k-1} **do**

    for all 1-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

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

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

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 string S = ACGATGTAC

	S_j	1	2	3	4	5	6	7	8	9
S_i		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-palindrome subsequences from U_1, there is no 1-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-palindrome subsequences from U_1, there is a 1-palindrome subsequence (3, 6) which satisfies that 3 > 1 and 6 < 8. We combine (1, 8) with (3, 6) to be 2-palindrome subsequence (1, 8) (3, 6) and add it into the set U_2.
U_2 = {(1, 8) (3, 6)}
There is another 1-palindrome subsequence (5, 7) which can satisfy that 5 > 1 and 7 < 8. We combine (1, 8) with (5, 7) to be 2-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-palindrome subsequence which can be satisfied.
U_2 = {(1, 8) (3, 6), (1, 8) (5, 7)}

(1-3)

We take the 1-palindrome subsequence (2, 9) from U_1.
There is a 1-palindrome subsequence (3, 6) which can be satisfied. We combine (2, 9) with (3, 6) to be 2-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-palindrome subsequence (4, 8) which can be satisfied. We combine (2, 9) with (4, 8) to be 2-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-palindrome subsequence (5, 7) which can be satisfied. We combine (2, 9) with (5, 7) to be 2-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-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-palindrome subsequence (3, 6) from U_1.
Check all 1-palindromes from U_1.
There is no 1-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-palindrome (4, 8) from U_1.
Check all 1-palindromes from U_1.
There is a 1-palindrome (5, 7) which can be satisfied. We combine (4, 8) with (5, 7) to be 2-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-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-palindrome (5, 7) from U_1.
Check all 1-palindromes from U_1.
There is no 1-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) (3, 6), (1, 8) (5, 7), (2, 9) (3, 6), (2, 9) (4, 8), (2, 9) (5, 7), (4, 8) (5, 7)}, U_3 = {}

(2-1)

We take the 2-palindrome (1, 8) (3, 6) from U_2.
Check all 1-palindrome from U_1.
There is no 1-palindrome which can be satisfied.
U_3 = {}

(2-2)

We take the 2-palindrome (1, 8) (5, 7) from U_2.
Check all 1-palindrome from U_1.
There is no 1-palindrome which can be satisfied.
U_3 = {}

(2-3)

We take the 2-palindrome (2, 9) (3, 6) from U_2.
Check all 1-palindrome from U_1.
There is no 1-palindrome which can be satisfied.
U_3 = {}

(2-4)

We take the 2-palindrome (2, 9) (4, 8) from U_2.
Check all 1-palindrome from U_1.
There is a 1-palindrome (5, 7) which can be satisfied. We combine (2, 9) (4, 8) with (5, 7) to be 3-palindrome (2, 9) (4, 8) (5, 7) and add it into the set U_3.
U_3 = {(2, 9) (4, 8) (5, 7)}

(2-5)

We take the 2-palindrome (2, 9) (5, 7) from U_2.
Check all 1-palindrome from U_1.

There is no 1-palindrome which can be satisfied.
U_3 = {(2, 9) (4, 8) (5, 7)}

(2-6)
We take the 2-palindrome (4, 8) (5, 7) from U_2.
Check all 1-palindrome from U_1.
There is no 1-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) (3, 6), (1, 8) (5, 7), (2, 9) (3, 6), (2, 9) (4, 8), (2, 9) (5, 7), (4, 8) (5, 7)}, U_3 = {(2, 9) (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-palindrome from U_1.
There is no 1-palindrome which can be satisfied.
U_4 = {}

Finally, we get the set $U = U_1 \cup U_2 \cup \ldots \cup U_{n/2}$ 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

5. Conclusions

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 biological characteristics. Our algorithm provides an effective tool for the related research.

References

[1] Allison, L. (2004) Finding Approximate Palindromes in Strings Quickly and Simply
[2] Choi, Charles Q (2005) DNA palindromes found in cancer. The Scientist
[3] Gusfied, D. (1997) Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, New York.
[4] Manacher, D. (1975) A new Linear-Time "On-Line" Algorithm for Finding the Smallest Initial Palindrome of a String. J. Assoc. Comput.
[5] Proto, A. H. L. and Barbosa V. C. (2002) Finding Approximate Palindromes in Strings. Pattern Recognition
[6] Tanaka, Hisashi; BERGSTROM, Donald A; YAO, Meng-Chao and TAPSCOTT, Stephen J (2006) Large DNA palindromes as a common form of structural chromosome aberrations in human cancers. Human Cell
[7] Wen, W. H. (2006) Longest Palindrome and Tandem Repeat Subsequences