Research on Pairwise Sequence Alignment Needleman-Wunsch Algorithm

Xiantao Jiang[1, a,*],Xueliang Fu[1, b],Gaifang Dong[1, c] and Honghui Li[1, d]

[1] College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot, 010018, P.R. China

[a]751822546@qq.com, [b*]fxliang@126.com,[c]949636417@qq.com, [d]lihh_chf@163.com

*corresponding author

Keywords: Bioinformatics,Pairwise sequence alignment, Needleman-Wunsch algorithm, Dynamic programming.

Abstract: The pairwise sequence alignment algorithm, Needleman-Wunsch is one of the most basic algorithms in biological information processing. However, the Needleman-Wunsch algorithm based on dynamic programming gets optimal alignment results with high time complexity and space complexity,which is impractical.This paper proposes an improved algorithm of Needleman-Wunsch, and demonstrates the algorithm by experiment. With the same score and accuracy, we compare and analyze the running time before and after improvement. The experimental results show that the improved algorithm can reduce the time complexity of Needleman-Wunsch algorithm.

1. Introduction

Sequence alignment is one of the core research fields of bioinformatics[1]. In biological studies, in order to determine whether two sequences are sufficiently similar[2] and determine whether they are homologous[3], sequence alignment is usually required. Sequence alignment can be divided into pairwise sequence alignment and multiple sequence alignment, according to the number of aligned sequences. And sequence alignment can also be divided into global alignment and local alignment, according to the range of aligned sequences.

At present, most of the pairwise sequence alignment algorithms are based on dynamic programming in operational research. The typical and widely used pairwise alignment algorithms are Needleman-Wunsch algorithm[4,5], Smith-Waterman algorithm[6,7,8], BLAST algorithm[9,10], FASTA algorithm[11,12] and Ukkonen algorithm[13]. Among the five algorithms, Smith-Waterman algorithm applies to two sequences that in relations are far and that are not similar as a whole, but has local similarity in some small area. But the Needleman-Wunsch algorithm applies to the analysis of the whole sequence with higher degree of similarity, which belongs to global alignment algorithm. The result of the algorithm is optimal, but it costs time and space complexity. However, the high complexity of the algorithm has become a bottleneck in the follow-up study. Therefore, if we can reduce the time and space complexity of the algorithm, it will surely contribute to the study of sequence alignment problem, which is the main purpose of this paper.

2. Method

This paper is divided into five parts. The first part mainly introduces related work on the pairwise sequence alignment, and describes the problems of Needleman-Wunsch algorithm. In the second part, we mainly describe the mathematic definition of the pairwise sequence alignment. The third part mainly describes the Needleman-Wunsch algorithm and puts forward the idea of improvement. The fourth part mainly verifies the improved Needleman-Wunsch algorithm based on experimental results. The fifth part is the summary of the whole paper,indicating that the improved Needleman-Wunsch algorithm is better than the original Needleman-Wunsch algorithm.

3. Mathematic Description of Pairwise Sequence Alignment

$x=x_1x_2\ldots x_n$ and $y=y_1y_2\ldots y_m$ are two sequences in a character set Σ. In the biological sequence, if x or y is a DNA sequence, Σ= {A, G, C, T}, consisting of four bases A, G , C, T. |x| denotes the length of the DNA sequence x. If x or y is a protein sequence, Σ= {A,R,…, V}, consisting of 20 amino acids. |x| denotes the length of the protein sequence x.

Definition1. If x and y are two random characters, σ(x,y), a score function, represents the comparison score of x and y. When x or y is null, we use "-" to describe it. In different algorithms, the score function is different.

Definition2. Given two sequences $x=x_1x_2\ldots x_n$ and $y=y_1y_2\ldots y_m$, we use |x| to represent the length of x, use x_i to represent the i-th character of sequence x. If x and y are the same, they must meet the following conditions:

(1) |x|=|y|;

(2) $x_i = y_i$, (0<i<=|x|);

Definition3. If x and y are two sequences, x and y after global alignment can be represented by sequences x' and y', including:

(1) |x'|=|y'|;

(2) Removing the empty characters in sequences x' and y', we can get sequences x and y;

For example: two sequences S= ATGGTAT and T= ATTGTCT, the result of the alignment is: S'= AT-GGTA-T; T'= ATTG-T-CT.The sequence alignment is to list sequences S' and T', and compare the corresponding position one by one. The final alignment scores of sequences S and T can be expressed by the following formula:

$$Score = \sum_{i=1}^{s'} \sigma\left(s'[i], T'[i]\right) \tag{1}$$

Definition4. For the two sequences x and y, the optimal global alignment sequence is the sequence with the highest score in all similarity alignments of x and y.

4. Needleman-Wunsch Algorithm

4.1 The Basic Idea of Needleman-Wunsch Algorithm

The basic idea of Needleman-Wunsch algorithm can be described as: use iterative method to calculate the similarity score of two sequences, and store the results in a score matrix. According to the score matrix, we can find the optimal alignment sequence by backtrace.

Assuming that two alignment sequences $SqA=s_1s_2\ldots s_n$ and $SqB=t_1t_2\ldots t_m$ constitute a two-dimensional matrix M. And n and m respectively represent the length of each sequence. s_i represents the i-th character of sequence SqA, t_j represents the j-th character of sequence SqB (1 <= i <= n, 1 <= j <= m), $M_{i,j}$ represents the optimal alignment score of the two sequences.

Take DNA sequence S= ACAGTAG and T= ACTCG as an example. The scoring rules are as follows:

$$\sigma(S_i, T_j) = \begin{cases} 1, S_i = T_j \\ 0, S_i \neq T_j \\ -1, S_i = \text{-或} T_j = \text{-} \end{cases}$$

Specific steps are as follows:

1. We should construct a (|S|+1) * (|T|+1)-order score matrix M and initialize M. As shown in Figure 1. The initial conditions are:

$$M(0,0) = 0$$
$$M(S_i,0) = M(S_{i-1},0) + \sigma(S_i,\text{-}), 1 \le S_i \le |S|$$
$$M(0,T_j) = M(0,T_{j-1}) + \sigma(\text{-},T_j), 1 \le T_j \le |T|$$

S_i \ T_j		A	C	T	C	G
	0	-1	-2	-3	-4	-5
A	-1					
C	-2					
A	-3					
G	-4					
T	-5					
A	-6					
G	-7					

Fig. 1 The initialization of score matrix M

2. Fill the score matrix. Calculate the score matrix iteratively by the following formula:

$$M_{i,j} = \max\begin{cases} M_{i-1,j-1} + \sigma(S_i, T_j) \\ M_{i-1,j} + \sigma('-', T_j) \\ M_{i,j-1} + \sigma(S_i, '-'). \end{cases} \quad (2)$$

In the score matrix, there are three paths to reach the position of $M_{i,j}$, as is shown in Figure 2:

① If the value of $M_{i,j}$ comes from the value of $M_{i-1,j-1}$ in the diagonal direction, there is no gap penalty, and the score is $\sigma(S_i,T_j)$.

② If the value of $M_{i,j}$ comes from the value of $M_{i-1,j}$ in the vertical direction, the gap penalty is $\sigma(S_i, -)$.

③ If the value of $M_{i,j}$ comes from the value of $M_{i,j-1}$ in the horizontal direction, the gap penalty is $\sigma(-, T_j)$.

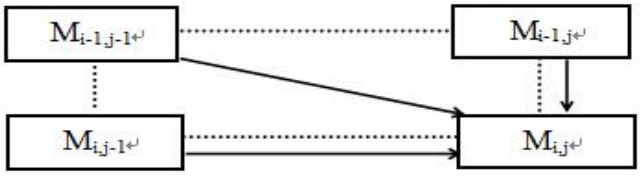

Fig. 2 The path of filling

The code of filling as follows:

```
for (i =1; i <=|S|; i++) {
    for (j =1; j <=|T|; j++){
        if(f1.charAt(i-1)==f2.charAt(j-1)){
            F[i][j].value=max(F[i-1][j-1].value+match,
            F[i-1][j].value+gap,F[i][j-1].value +gap);
        }else{
            F[i][j].value=max(F[i-1][j-1].value+dismatch,
            F[i-1][j].value+gap,F[i][j-1].value +gap);
        }
    }
}
```

In the penalty rule, the word "match" indicates the score when the corresponding characters are same, and the word "dismatch" indicates the score when the corresponding characters are not same,and the word "gap" indicates the penalty of inserting a space.

3. Processing backtrack from the lower right corner to the upper left corner of the score matrix. Finally, we can obtain the optimal global alignment results. The path of backtrack is shown in figure 3:

S_i \ T_j		A	C	T	C	G
	0	-1	-2	-3	-4	-5
A	-1	1	0	-1	-2	-3
C	-2	0	2	1	0	-1
A	-3	-1	1	2	1	0
G	-4	-2	0	1	2	2
T	-5	-3	-1	1	1	2
A	-6	-4	-2	0	1	1
G	-7	-5	-3	-1	0	2

Fig. 3 The path of backtrack

Based on the above steps, we can find that the time complexity and space complexity of Needleman-Wunsch algorithm are O (| S | * | T |).

4.2 The Improvement of Needleman-Wunsch Algorithm

Since the most time-consuming process is filling the score matrix, this paper mainly improve filling method of the score matrix.The specific idea of the improved Needleman-Wunsch algorithm: In the filling of the score matrix, the score of current value $M_{i,j}$ is obtained by recursive formula, which is:

$$M_{i,j} = \max\begin{cases} M_{i-1,j-1} + \sigma(S_i, T_j) \\ M_{i-1,j} + \sigma('-', T_j) \\ M_{i,j-1} + \sigma(S_i, '-'). \end{cases}$$

It can be seen from the formula that there are three scores of diagonal direction, vertical direction and the horizontal direction. By adding penalty point respectively, we can get three scores, of which the maximum value is the current value of $M_{i,j}$.That is to say, the current value of $M_{i,j}$ needs to be calculated by max function.

In sequences $S=s_1s_2\ldots s_n$ and $T=t_1t_2\ldots t_m$, if $S_i=T_j$ (S_i represents the i-th character of sequence S, T_j represents the j-th character of sequence T.), $M_{i,j}$ is the score in diagonal direction adding penalty point, without calculating the score in vertical direction and the horizontal direction. Therefore, if $S_i=T_j$, the current value of $M_{i,j}$ doesn't need to be calculated by the max function,reducing the running time and improving efficiency.

The improved filling code is as follows:

```
for (i = 1; i <=f1.length(); i++) {
    for (j =1; j <=f2.length(); j++){
        if(f1.charAt(i-1)==f2.charAt(j-1)){
            F[i][j].value=F[i-1][j-1].value+match;
        }else{
            F[i][j].value=max(F[i-1][j-1].value+dismatch,
             F[i-1][j].value +gap,F[i][j-1].value +gap);
        }
    }
}
```

5. Experiment and Analysis of the Algorithm

The experimental server is Lenovo, Wanquan R680 G7. Operating system is Linux. Development tool is Java 1.7. Hardware environment: CPU is Xeon E7-4820, and memory is 1TB.

The experimental data is from the NCBI website. The following experiment is to compare the similarity of Sichuan snub-nosed monkey and Yunnan snub-nosed monkey. As is shown in Table 1, the Rhinopithecus bieti in the sequence 1 represents the length of the sequence of Yunnan snub-nosed monkey, and the Rhinopithecus roxellana in the sequence 2 represents the length of the

sequence of Sichuan snub-nosed monkey. For example, the Rhinopithecus bieti-734 represents that the length of the sequence of Yunnan snub-nosed monkey is 734bp, and the Rhinopithecus roxellana-702 represents that the length of the sequence of Sichuan snub-nosed monkey is 702bp. The two sequences are read into the program to test in the form of text, so as to obtain the similarity, the final score and the consuming time.

Table 1 The comparing results of the improved and original Needleman-Wunsch algorithm

Length of sequence 1	Length of sequence 2	The similarity of original algorithm %	The similarity of improved algorithm %	The score of the original algorithm	The score of the improved algorithm	The running time of the original algorithm 1 (ms)	The running time of the improved algorithm 1 (ms)	The running time of the original algorithm 2 (ms)	The running time of the improved algorithm 2 (ms)	The running time of the original algorithm 3 (ms)	The running time of the improved algorithm 3 (ms)
Rhinopithecus bieti-734	Rhinopithecus roxellana-702	80.22%	80.22%	546	546	40	36	36	36	42	36
Rhinopithecus bieti-849	Rhinopithecus roxellana-845	99.06%	99.06%	837	837	46	43	40	40	39	38
Rhinopithecus bieti-2377	Rhinopithecus roxellana-2290	94.16%	94.16%	2129	2129	155	110	152	112	151	111
Rhinopithecus bieti-3795	Rhinopithecus roxellana-3489	87.92%	87.92%	2978	2978	335	246	345	240	330	246
Rhinopithecus bieti-4776	Rhinopithecus roxellana-4743	95.65%	95.655	4452	4452	582	380	584	388	569	395
Rhinopithecus bieti-5543	Rhinopithecus roxellana-5548	99.80%	99.80%	5532	5532	747	513	792	518	791	516
Rhinopithecus bieti-6932	Rhinopithecus roxellana-7378	88.51%	88.51%	5860	5860	1287	942	1290	908	1296	954
Rhinopithecus bieti-10377	Rhinopithecus roxellana-10130	97.24%	97.24%	9839	9839	2698	2044	2785	2135	2779	2116
Rhinopithecus bieti-15101	Rhinopithecus roxellana-15043	97.53%	97.53%	14610	14610	5955	4204	5732	4325	5860	4139

The experimental results (Table 1) show that the matching rates of the original and improved Needleman-Wunsch algorithm are the same. The scores of the original and improved Needleman-Wunsch algorithm are also the same. That is to say, the similarity of two sequences is the same. In figure 4, for the consuming time, we get the average value of sequence alignment of three times. And the time consumed by the improved Needleman-Wunsch algorithm is generally shorter than that of the original Needleman-Wunsch algorithm.

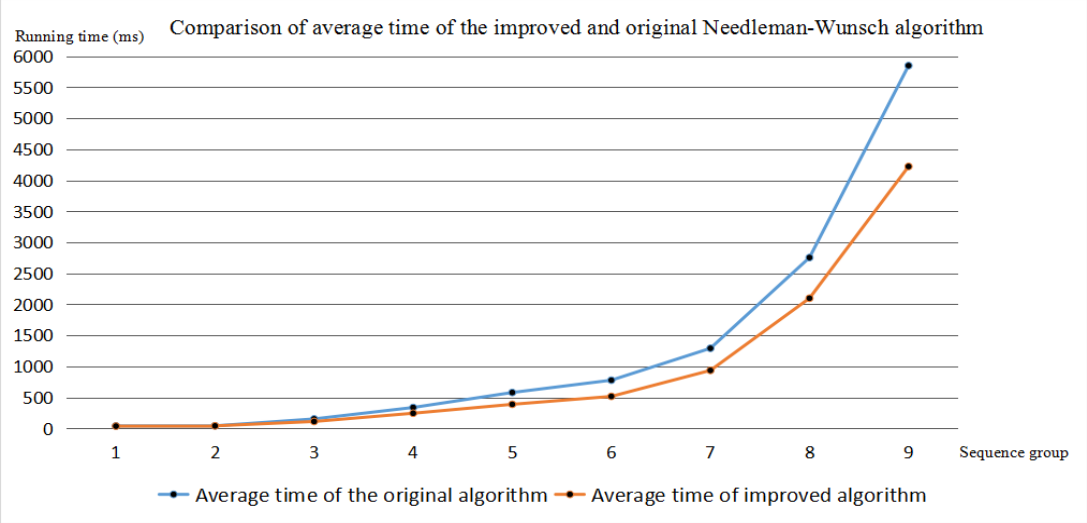

Fig. 4 Comparison of average time of the improved and original Needleman-Wunsch algorithm

6. Summary

The sequence alignment is one of the most fundamental problems in biological information processing, and it is an important part of bioinformatics. The purpose of sequence alignment is to determine the degree of similarity between genes, and determine the homology between the sequences.

This paper deeply studied the Needleman-Wunsch algorithm, and discussed the research status and the basic idea of Needleman-Wunsch algorithm in detail. Meanwhile, the improved idea of this algorithm is presented. By analyzing the experimental data, the improved algorithm can effectively reduce the time complexity of biological sequences alignment.

7. Acknowledgments

This research was financially supported by Chinese Natural Science Foundations (61363016, 61063004), Key Project of Inner Mongolia Advanced Science Research (NJZZ14100), Inner Mongolia Colleges and Universities Education Department Science Research (NJZC059), Natural Science Foundation of Inner Mongolia Autonomous Region of China (NO.2015MS0605, NO.2015MS0626 and NO.2015MS0627) and Ministry of Education Scientific research foundation for Study abroad personnel[2014] 1685.

References

[1] Zhang Chunting, the Current Status and Prospect of Bioinformatics, the Research and Development of World Science and Technology, 2000, 22(6):1720.

[2] T.K. Attwood, D.J. Parry-Smith. Introduction to Bioinformatics. Luo Jingchu. Beijing: Peking University Press, 2002.

[3] Minoru Kanehisa. Post-genome Informatics. Sun Zhirong translation. Beijing: Tsinghua University Press, 2002.

[4] S Needleman,C Wunsch. A general method applicable to the search for similarities in the amino acid sequences of two proteins.Journal of Molecular Biology, 1970,48: 443-453

[5] D Sankoff. Matching sequences under deletion/insertion constraints. Proc Natl Acad Sci.USA, 1972, 69:4-6.

[6] Zhong Yang, Zhang Liang, Zhao Qiong. Brief bioinformatics. Beijing: Higher Education Press, 2001.

[7] T Smith,M Waterman. Identification of common molecular sequence.Journal of Molecular Biology, 1981, 147:195-197.

[8] W Goad, M Kanehisa. Petern recognition in nucleic acid sequences. A General method for finding local homologies and symmetries.Nucleic Acids Research, 1982, 10(1):247-263.

[9] S Altschul, W Gish,W Miller.Basic local alignment search tool. Journal of Molecular Biology, 1990, 215:403-410.

[10] S Altschul,T Madden, AA Schaaefer. Gapped BLAST and PSIBLAST:a new generation of protein database search programs.Nucleic Acids Research, 1997,25(17):3389-3402.

[11] W Wilbur, D Lipman. Rapid similarity searches of nucleic acid and protein data banks.Proc Natl Acad Sci.USA,1983,80:726-730.

[12] D Lipman, W Pearson. Rapid and sensitive protein similarity searches. Science, 1985, 227:1435-1441.

[13]Tang Yurong. Study on Sequence Alignment Algorithm in Bioinformatics. China Agricultural University Doctoral Dissertations.2004.8-20.