A Pairwise Alignment Algorithm for Long Sequences of High Similarity

Chien-Tai Lee and Sheng-Lung Peng

Abstract Alignment algorithms are important in bioinformatics for comparing the similarity among sequences. The algorithm of Needleman–Wunsch is well known for globally aligning two sequences. However, this algorithm is unsuitable for sequences of long length. Many heuristic algorithms are proposed, such as BLAST and FASTA. However, they are still unsuitable for long sequences. In this paper, we study the alignment problem on highly similar sequences. By taking SARS viruses as an example, our result shows that our algorithm runs faster than Clustalx for aligning two SARS viruses. It implies that our algorithm is suitable for viruses of high similarity.

Keywords Pairwise alignment · SARS virus · Whole-genome alignment

1 Introduction

Sequence alignment is an important technique for comparing the similarity of two biological sequences in bioinformatics. In 1970, biologists Needleman and Wunsch were the first to analyze biological hereditary information by using electronic calculator. They applied the method of dynamic programming to analyze the similarity between two amino acid sequences [1]. In 1988, the US government established the National Center for Biotechnology Information (NCBI) to help in creating automatic analyzing system of biological information. This further enhances the ability for biomedical researchers to search and analyze biological information more efficiently and accurately. In 1992, the most famous nuclear acid sequence database, GenBank, began the service for the whole world. It collected more than 150 billion of base pairs counted to February 2013. Meanwhile, in Europe, the European Molecular Biology Laboratory (EMBL) was founded in
DNA is the most important element in the study of life science. After knowing the sequences of DNA and RNA, the next important thing is to analyze all the information originated from the DNA sequences. There are three billion of bases to form chromosomes in human beings. It would take tremendous time to know all the sequences in human genes. It could be more difficult to analyze the underlying meaning in the DNA sequences. Recently, the rapid development in information science and technology has resolved a lot of problems. It has become more popular to combine information technology and biological science, thereby trying to have a big breakthrough in the fields of biomedicine and pharmaceutical research. With the help of computer technology, this goal may be expected in the future.

The main subjects in biological information science include prediction of a protein structure, comparison of the similarity and the homology of sequences, phylogeny construction, genome sequence analysis. In this paper, we study the DNA strains of Severe Acute Respiratory Syndrome (SARS) viruses. We propose an algorithm for aligning two highly similar sequences. In our experimental result, if these two sequences are highly similar, the expected time for alignment is almost linear.

### 2 Preliminaries

A virus is a simplest microorganism. All its hereditary information is located on the chains of nucleotides. Once something is changed in the nucleotide chain, it results in a variation in the descendant. Further, viral gene underwent spontaneous mutation during the process of proliferation. Although most of them died, some survived and acquired the ability to adjust themselves to the stress from the surrounding environment.

A DNA mutation has many types, e.g., insertion, deletion, substitution (replacement), duplication. Mutations can be defined in two ways:

1. Chromosomal mutations: They are modifications of a chromosome. For example, in the deletion or duplication of genes (or segments) of a chromosome, it changes the total number of chromosomes sometimes.

2. Gene mutations: These mutations occur in the nucleotide sequence and are caused by the substitution of one nucleotide for another or by insertion or deletion of one or more nucleotides in the DNA. More specifically, insertion means addition of one or many nucleotides from DNA. Deletion means removal of one or many nucleotides from DNA. Substitution (replacement) means to replace one DNA base by another DNA base.
There are two popular methods available to estimate the similarity of two sequences. One applies the principle of probability and statistics [2], and the other is to use a sequence alignment algorithm. Usually, we use indel to refer deletions or insertions of an alignment.

2.1 Global Alignment

Dynamic programming is an important technique for algorithm design. It usually solved an optimization problem by caching subproblem solutions rather than recomputing them [3–5]. Needleman–Wunsch’s algorithm [1] uses dynamic programming technique to compute an optimal alignment for two sequences. It uses a two-dimensional array $D$ to store the best scores for each entry. Therefore, they can guarantee to find an optimal solution, but consume resources very much, and it is not efficient for comparing with long sequences. This method has two main steps:

1. Compute similarity scores

- A score is computed for each entry in the array according to their similarity.
- The similarity score is usually defined as [6, 7]:

$$D(i,j) = \max \left( \frac{D(i-1,j-1) + \delta(a_i,b_j)}{D(i-1,j) + \delta(a_i,-)}, \frac{D(i,j-1) + \delta(-,b_j)}{D(i,j-1)} \right).$$

2. Construct an alignment

- Backtrack the matrix to obtain an optimal alignment.

In the optimization of the alignment algorithm, we may need to add some gaps to the sequence. The frequency of “insertion” or “deletion” events causes gap insertions. Thus, we can use different penalty parameters to define a gap insertion.

In Needleman–Wunsch’s algorithm, we need to store $(n + 1) (m + 1)$ numbers for aligning a sequence of length $n$ with another sequence of length $m$. By the above-mentioned formula, each number takes a constant number of computations. Thus, the algorithm runs in $O(nm)$ time and requires $O(nm)$ memory. For whole-genome alignment, this running time is still too slow. In other words, it is not feasible for real biological applications.
2.2 Heuristic Algorithms

Generally speaking, aligning two sequences with length at most \( n \), the time complexity is \( O(n^2) \). This is by no means the best time bound. However, when aligning three sequences, it consumes longer execution time, e.g., \( O(n^3) \). How to reduce the time complexity is the most important work while performing the whole-genome alignment. Many heuristic methods are proposed. The kind of these methods is an algorithm that usually but not always gives an optimal answer.

For example, to use the most famous tools such as FASTA or BLAST (Basic Local Alignment Search Tool) programs to speed up the performance, the problem is to compromise its sensitivity. Therefore, they do not guarantee to find an optimal solution, but can improve the efficiency.

- FASTA series [8]: The method of these series is quite precise, but the speed is slower. It can only compare a nucleic acid sequence to the nucleic acid information bank, or compare a protein sequence to the protein information bank [8–11].
- BLAST series [12]: The method of these series has a very quick search speed but has the fault when the sequences have low similarity. It contains a group of programs and can automatically execute according to the information bank type of input sequences [12, 13].

2.3 Genome-Scale Alignment: MUMmer

MUMmer is a tool for aligning entire genomes [14]. It has three main steps:

1. Computing MUMs: A MUM (maximal unique match) for two sequences \( x \) and \( y \) is a pair of subsequences \( (x', y') \), that is, an exact match, and there is no other matching subsequence pair containing \( x' \) and \( y' \) simultaneously. During the computation, MUMmer first constructs a suffix tree for sequence \( x \). Then, the suffixes of \( y \) are inserted to the constructed tree. It is called a generalized suffix tree. Finally, all the MUMs can be computed by traversing this generalized suffix tree.

2. Finding the backbone of the alignment: All the MUMs in \( x \) are arranged in an increasing order. Then, we find a longest increasing subsequence of MUMs for \( y \) with respect to the MUMs of \( x \). These MUMs define the backbone of the alignment.

3. Closing gaps: To obtain a final result, the gaps between consecutive MUMs of the backbone are aligned by using the Smith–Waterman algorithm [15].

MUMmer is also suitable for two sequences with high similarity. However, it has two flaws: One is that it needs a very large space to save the suffix tree, and the
other is that MUMs cannot be found when the same substring appeared more than once. Recently, by using the computational power of GPUs, an algorithm for very long sequence is proposed [16].

3 The Proposed Algorithm

In bioinformatics, pairwise sequence alignment is the most important tool. Many tools use it as a core for approximating the multiple sequence alignment or comparing the similarity of two genes. The Needleman–Wunsch alignment is the best pairwise sequence alignment algorithm for finding an optimal solution. It uses a dynamic programming approach and runs in $O(mn)$ time where $m$ and $n$ are the lengths of the two input sequences, respectively. However, while considering the whole genome as an input, this algorithm is impractical. Many heuristic algorithms occur for aligning two whole-genome sequences. For example, MUMmer uses the maximal unique matching sequences as a base for aligning two entire genomes. MUMmer uses the data structure of suffix tree to find all the maximal unique matching sequences. Although genomes have high similarity, it still needs a very large space. Thus, we propose a new method for saving space and time to align two entire highly similar sequences.

SARS virus, one kind of coronavirus, is a highly transmissible and virulent virus that caused a disastrous outbreak in the Southeast Asia in 2003. Many scientists and physicians wanted to know more about different strains of SRAS viruses and tried their best to prevent the spread of this disease. Its DNA sequence is approximately 29,700 base pairs. Here, we propose an algorithm to do the alignments for SARS viruses. Our algorithm can be divided into three phases.

1. The first phase is Dot Matrix Phase: We use dot matrix to find the longest common substring (LCS) in a sliding window. The segment in sliding window is broken into three parts by the LCS: the LCS itself, prefix string that before the LCS, and the suffix string that after the LCS.
2. The second phase is Needleman–Wunsch Phase: Let the prefix strings be aligned by the Needleman–Wunsch’s algorithm, the follow-up for getting optimal alignment with appending the LCS.
3. The third phase is the Liner Comparison Phase: If there is only one remaining suffix string, then we compare it with the other uncompared string. If the two sequences are highly similar, then we obtain a linear alignment.

We define three variables SW, SR, and MES as follows:

- Sliding window (SW): determination of a length for Dot Matrix Phase.
- Similarity rate threshold (SR): a threshold of similarity ratio for the two sub-sequences in sliding window.
- Max continuous equal substring length threshold (MES): a threshold for the length of LCS in sliding window.
First, let $S_1$ and $S_2$ be two sequences for doing an alignment. From the beginning of the two sequences, determine a sliding window size $SW$ to perform dot matrix. An example is shown in Fig. 1. The plots in dot matrix provide an easy and efficient way to find similarity between two sequences.

The alignment should be a diagonal of continuous dots. Sometimes, it is broken at a point of mutation and shifting to another diagonal because of indels. In this method, we find the LCS and determine whether the similarity is greater than $SR$ and the length of LCS is greater than $MES$. If the answer is no, then we double the window size. We then repeat the procedures until one of the previous two conditions is satisfiable. Then, we do an alignment using Needleman–Wunsch’s algorithm on the prefix strings.

After finishing a partial alignment, we then extend the alignment from LCS by comparing it with the follow-up sequences until they are different. Finally, we repeat the above procedure for the remaining subsequences. Figure 2 shows the concept of a phase of our algorithm.

Fig. 1 Dot matrix

Fig. 2 Complete alignment in sliding window
For any two sequences with high similarity, the alignment is very efficient because the linear comparison does a favor. However, this method is only suitable for highly similar sequences.

4 Experimental Results

For obtaining a quantitative comparison, we use an old computer system to run the proposed algorithm. Our experimental environment is as follows:

- Machine: Compaq
- CPU: Intel® Xeon™ CPU 3.06 GHz (2CPUs)
- Memory: 1024 MB
- Operating system: Microsoft Windows 2000

We use 102 SARS sequences of FASTA format as an input (which are downloaded from NCBI). The FASTA format begins with a single-line description and is followed by lines of sequence data. The description line starts with “>” symbol for distinguishing from the other sequence data. It is recommended that each line of text should be shorter than 80 characters. An example of FASTA format is given as follows:

```
>gi|31416306|gb|AY279354.2|SARS coronavirus BJ04
TACCCAGGAAAAGCCAACCAACCTCGATCTCTTGTAGATCTGTTCTCTAAACGAACTTTAAAATCTGTGT
AGCTGTGCTCGGCTGCAATGTGCTGACCTACCTGACATCAGCTGACATCAGCTGACATCAGCTGAC
........
CGGCCACAAGGTGTGGTACGGTGCAATGGTGCAATGGTGCAATGGTGCAATGGTGCAATGGTGCA
ATAAGGGAGCCGGTGGTACGGTGCAATGGTGCAATGGTGCAATGGTGCAATGGTGCAATGGTGCA
```

A SARS of whole genome has about 29,700 base pairs. We use our algorithm to align and analyze similar degree according to the matching rules of IUPAC (International Union of Pure and Applied Chemistry).

Clustalx is famous software in bioinformatics. It performs alignments and calculates distances for all pairs of sequences. We randomly take out 10 sequences from 102 SARS sequences to do a comparison with SARS NC004718.3. Table 1 shows the results.
| Time in mm:ss | SinP5 | Sin852 | TW4  | LC3  | HZS2-E | Shang | QXC2 | PUMC 03 | TWC3  | TWK | RNA | BJ02 |
|---------------|-------|--------|------|------|--------|-------|------|--------|-------|-----|-----|------|
| Clustalx 1.83 | 9:49  | 9:49   | 9:50 | 9:42 | 9:36   | 9:50  | 9:50 | 9:50   | 9:50  | 9:50| 9:90|
| Our method    | 0:02  | 0:18   | 0:01 | 4:36 | 0:01   | 5:19  | 0:02 | 0:01   | 0:01  | 0:01| 0:02|
| Seq. length   | 29,713| 29,670 | 29,729| 29,350| 29,736 | 29,013| 29,745| 29,727 | 29,727| 29,745|      |

Table 1 A comparison with Clustalx
5 Conclusions

Bioinformatics has become the key to answer the mystery of life in the post-genomic era. This paper proposes a sequence alignment algorithm for those DNAs of high similarity. Although the Needleman–Wunsch alignment is the best pairwise sequence alignment algorithm for finding an optimal solution, it is impractical for whole genomes. MUMmer’s algorithm needs a very large extra space for saving the suffix tree, and it is also time-consuming for searching MUMs in the suffix tree. We provide another kind of alignment algorithms. It cannot avoid the limitation of $O(n^2)$, but it has a good performance for highly similar sequences.

Acknowledgements This work was partially supported by the Ministry of Science and Technology of Taiwan, under Contract No. MOST 104-2221-E-259-006.

References

1. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequences of two proteins. J. Mol. Biol. 147, 195–197 (1970)
2. Smith, T.F., Waterman, M.S., Burks, C.: The statistical distribution of nucleic acid similarities. Nucleic Acids Res. 13, 645–656 (1985)
3. Chao, K.M., Pearson, W.R., Miller, W.: Aligning two sequences within a specified diagonal band. Comput. Appl. Biosci. 8, 481–487 (1992)
4. Smith, T.F., Waterman, M.S., Burks, C.: The statistical distribution of nucleic acid similarities. Nucleic Acids Res. 13, 645–656 (1985)
5. Pearson, W.R., Miller, W.: Dynamic programming algorithms for biological sequence comparison. Methods Enzymol. 210, 575–601 (1992)
6. Fitch, W.M., Smith, T.F.: Optimal sequence alignments. Proc. Natl. Acad. Sci. USA 80, 1382–1386 (1983)
7. Gotoh, O.: An improved algorithm for matching biological sequences. J. Mol. Biol. 162, 705–708 (1982)
8. Pearson, W.R., Lipman, D.J.: Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. USA 85, 2444–2448 (1988)
9. Fickett, J.W.: Fast Optimal alignment. Nucleic Acids Res. 12, 175–180 (1984)
10. Pearson, W.R.: Rapid and sensitive comparison with FASTP and FASTA. Methods Enzymol. 183, 63–98 (1990)
11. Pearson, W.R.: Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics 11, 635–650 (1991)
12. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic Local Alignment search Tool. J. Mol. Biol. 215, 403–410 (1990)
13. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–402 (1997)
14. Delcher, A.L., Kasif, S., Fleishmann, R.D., Peterson, J., White, O., Salzberg, S.L.: Alignment of whole genomes. Nuc. Acids. Res. 27, 2369–2376 (1999)
15. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. 147, 195–197 (1981)
16. Li, J., Ranka, S., Sahni, S.: Pairwise sequence alignment for very long sequences on GPUs. Int. J. Bioinform. Res. Appl. 10(4–5), 345–368 (2014)