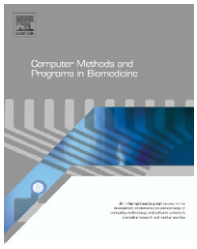
COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 86 (2007) 17–20

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



SGA: A grammar-based alignment algorithm

Guangyue Hu[a,*], *Shiyi Shen*[a], *Jishou Ruan*[a,b]

[a] *College of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, PR China*
[b] *Chern Institute of Mathematics, Nankai University, Tianjin 300071, PR China*

ARTICLE INFO

Article history:
Received 26 April 2006
Received in revised form
18 November 2006
Accepted 21 December 2006

Keywords:
Yang–Keiffer algorithm
Sequence similarity
Sequence alignment
Super pairwise alignment
Super genome alignment

ABSTRACT

As the cost of genome sequencing continues to drop, comparison of large sequences becomes tantamount to our understanding of evolution and gene function. Rapid genome alignment stands to play a fundamental role in furthering biological understanding. In 2002, a fast algorithm based on statistical estimation called super pairwise alignment (SPA) was developed by Shen et al. The method was proved to be much faster than traditional dynamic programming algorithms, while it suffered small drop in accuracy. In this paper, we propose a new method based on SPA that target analysis of large-scale genomes. The new method, named super genome alignment (SGA), applies Yang–Keiffer coding theory to alignment and results in a grammar-based algorithm. SGA has the same computational complexity as its predecessor SPA, and it can process large-scale genomes. SGA is tested by using numerous pairs of microbial and eukaryotic genomes, which serve as the benchmark to compare it with the competing BLASTZ method. When compared with BLASTZ, the result shows that SGA is significantly faster by at least an order of magnitude (for some genome pairs the differences is as large at two orders of magnitude), and suffers on average only about 1% loss of the similarity of alignment.

1. Introduction

With the development of sequencing techniques, alignment becomes a popular way to understand the evolution and function of genomes. Fast alignment for genomes plays an important role in computational biology and bioinformatics. Neddleman–Wunsch algorithm [1] and Smith–Waterman algorithm [2] are both dynamic programming methods with the complexity of $O(n^2)$, where n is the average length of input sequences. In 2002, Shen et al. developed super pairwise alignment (SPA) method [3], which is based on statistical estimation and its computation complexity is only $O(n)$. SPA is much faster than the Neddleman–Wunsch and the Smith–Waterman algorithms, but it also suffers small drop in accuracy. Recently, some other efficient methods have been developed, such as WU-BLAST [4], BLAT [5] and BLASTZ [6]. These are hybrid methods, which require a hash table of sequences to find seeds and then extend the seeds to find better alignments. In contrast, MUMmer [7], AVID [8] and LAGAN [9] are tree-based method, which take advantage of tree-structures to identify anchors and then align the substrings between anchors to finish the global alignment [10]. The above methods are capable of aligning a number of genomes with good accuracy, but only if the size of the query genome is not too long. Due to computational complexity, they will not provide results for super-size genomes with tolerable memory in reasonable time, such as *Arabidopsis thaliana* and mammalian. In this paper, we propose a grammar-based algorithm that aims to provide scalable solution for the super-size genomes. The proposed method is called super genome alignment (SGA), which is based on Yang–Kiffier algorithm [11–13] and our recent SPA algorithm.

* *Corresponding author*. Tel.: +86 2223505434.
E-mail address: huguangyue@gmail.com (G. Hu).
0169-2607/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2006.12.007

The outline of this paper is stated as follows: Firstly, we introduce the Yang–Kiffier and SPA methods. Secondly, we describe the proposed SGA method, and perform experiments to test its quality. Finally, we collect several large microbial and mammalian genomes as the benchmark set and compared SGA with other methods using these data.

2. Methods

Before describing the proposed alignment method, we first introduce three methods that are used to implement SGA, namely Yang–Keiffer algorithm [11] and SPA algorithm [3] and modulus structure [14–15].

2.1. The Yang–Keiffer (YK) algorithm

The *Yang–Keiffer* (YK) *algorithm* is a universal lossless data compression algorithm, which can asymptotically achieve the entropy rate of any stationary and ergodic source. The worst case redundancies among sequences of length n of YK code have an upper-bound of $c\log(\log n)/\log n$.

For instance, let x = aatcaatgcaatatat be the query sequence, according to YK algorithm, this sequence can be compressed as follows:

1. Let A_1 = aat, then the sequence is compressed as A_1cA$_1$gcA$_1$atat.
2. Let A_2 = cA$_1$ and A_3 = at, then the sequence is compressed as A_1A_2g$A_2A_3A_3$ and A_1 = aA_3, A_2 = cA$_1$and A_3 = at.
3. If we denote $S = A_1A_2$g$A_2A_3A_3$, then the formation of x = aatcaatgcaatatat is included in $V = \{S, A_1, A_2, A_3\}$, which is called the variable set. $S \to A_1A_2$g$A_2A_3A_3$, $A_1 \to$ aA_3, $A_2 \to$ cA$_1$ and $A_3 \to$ at are called the set of rules. In fact, replacing A_i into S, x can be recovered. The collection of $V = \{S, A_1, A_2, A_3\}$, $S \to A_1A_2$g$A_2A_3A_3$, $A_1 \to$ aA_3, $A_2 \to$ cA$_1$ and $A_3 \to$ at is called the grammar structure of the sequence and is denoted by G_x. For detail, the reader is referred to [11] and [12].

2.2. SPA algorithm

Biological mutations can be generally classified into four types: substitutions, transpositions, insertions and deletions. The first two types of mutations result in errors and the last two types of mutations change the lengths of input DNA (RNA) sequences. SPA algorithm combines the method of statistical estimation and combination analysis to deal with the last two types of mutations between two strings. DNA or RNA sequences can be considered as independently and identically distributed sequences of random variables. Based on a statistical model, SPA predicts the presence of insertions or deletions, and the length of insertions or deletions according to the local similarity of input sequences. In this way, all of the insertions or deletions and the lengths of gaps in both DNA (RNA) sequences will be scanned perfectly, and the computational complexity of SPA is $O(n)$, where n is the average length of input sequences. When all insertions or deletions have been found, input sequences are aligned and substitutions and transpositions can also be found.

2.3. Modulus structure

The *modulus structure* is a mathematical concept that was proposed to simplify the process of the alignment of DNA sequences. For any DNA sequence x, its modulus structure denoted by H_x, is defined as set $H_x = \{(i_{x,k}, l_{x,k})\}_{k=1}^{k_x}$ where k_x is the amount of gaps occurring in x, $i_{x,k}$ is the starting position of the kth gap, and $i_{x,k}$ is the length of the kth gap.

For instance, given the following three sequences:

- A_1 = gttagaagaagccaacaaaggagagaac.
- A_2 = acaggaagaagccaacaggagagaac.
- A_3 = cagttagagccaacaaaggagagaac.

And their multiple sequence alignment

- C_1 = —gttagaagaagccaacaaaggagagaac.
- C_2 = acag—gaagaagccaa–caggagagaac.
- C_3 = -cagttag—-agccaacaaaggagagaac.

The modulus structures are $H_{C_1} = \{(1, 3)\}$, $H_{C_2} = \{(5, 3), (16, 2)\}$ and $H_{C_3} = \{(1, 1), (8, 4)\}$, respectively.

2.4. The SGA algorithm

Most existing alignment methods for large-scale genomes are either hash-table-based or tree-based. In contrast, SGA method is grammar-based. It employs the inherent characteristics of the YK-code to convert a super long sequence into a shorter sequence and provides the grammar structure of input sequences. The proposed SGA algorithm consists of the following six steps:

- *Step 1*. Encode sequence x_1 by using YK-code to get its grammar structure G_1.
- *Step 2*. Based on G_1, encode sequence x_2 to get the joint grammar structure G of x_1 and x_2.
- *Step 3*. Based on the joint grammar structure G, we use V to denote the set of variables. If a variable A_i occurs in V, then the string corresponds to A_i will occur in both x_1 and x_2 at least once. Given that we denote $begin_1$ and $begin_2$ as the starting positions of the first string that occurs in x_1 and x_2, respectively, and l as the length of the string, we obtain triplet ($begin_1$, $begin_2$, l). For convenience, we say that the same string in different sequences is a grammatical match.
- *Step 4*. Rearrange all grammatical matches based on the information of $begin_1$ or $begin_2$. If the non-overlapping grammatical matches in two substrings of the two input sequences are sufficiently many so that the similarity of the two substrings is greater than a threshold, then these regions will be aligned by using SPA algorithm and be referred to as the grammatical blocks.
- *Step 5*. Align the gaps between anchors by using modulus structure and SPA.
- *Step 6*. Inverse x_2 and repeat steps 1–5 to find the inversions.

Preprocessing helps to improve the accuracy and the speed if the length of the sequence is large. After encoding a DNA sequence by using YK-code, each variable of its grammar structure corresponds to a repeated pattern. It is easy to mask repeats before alignment. Furthermore, biological features of