Fast global sequence alignment technique

Article *in* Circuits, Systems and Computers, 1977. Conference Record. 1977 11th Asilomar Conference on · November 2011

DOI: 10.1109/ACSSC.2011.6190171

2 authors:

Talal Bonny
University of Sharjah
30 PUBLICATIONS **98** CITATIONS

SEE PROFILE

Khaled Nabil Salama
King Abdullah University of Science and Technology
254 PUBLICATIONS **3,304** CITATIONS

SEE PROFILE

FAST GLOBAL SEQUENCE ALIGNMENT ALGORITHM

Talal Bonny and Khaled N. Salama

Electrical Engineering Program
King Abdullah University of Science and Technology (KAUST)
Thuwal, Kingdom of Saudi Arabia
Email: {talal.bonny, khaled.salama}@kaust.edu.sa

A sequence alignment tool is one of the most important tools in Bioinformatics. This tool needs to process large amounts of data, which is growing exponentially (may reach thousands of database sequences), and therefore it may take hours of mainframe time to get the optimum solution. For example, the Needleman-Wunsch [1] (for global alignment), which is implemented in [2], provides optimal alignment in a time that is proportional to the product of the lengths of the two sequences being compared.

One solution for this problem is to use heuristic alignment algorithms, which prune the search space by using fast approximate methods to locate the similarity region.

One of the well-known algorithms used for global sequence alignment is GAP [4, 5]. This algorithm provides global alignment of two sequences but its score of alignment is far from the optimal one when it is compared with the score of the global optimal alignment algorithm "Needleman-Wunsch".

In this work, we introduce our new and fast alignment method, which is called ABS (Alignment By Scanning), to provide an approximate alignment of any two DNA sequences independent from their similarities.

The alignment score of the ABS is better (closer to the optimal) than the GAP. In addition to that, it is much faster than the GAP and the Needleman-Wunsch.

The ABS Algorithm passes trough these three steps (see Fig. 1):

1- Finding the positions at which the two sequences, S and T, are split into different subsequences. We call these positions "Barriers" (step 1 in Fig. 1).

To find the barriers, the ABS scans the sequences S and T from the beginning toward the end. When a match is met, the ABS checks the relation between the letters of the sequence S before the match and the letters of the sequence T after the match and vice versa. If there is no relation, the match is considered to be a barrier and the sequences S and T may be split at that position. Otherwise, the ABS ignores the match and it continues scanning the sequences.

2- Aligning each subsequence separately, i.e., independent from the other (step 2 in Fig. 1).

The alignment is done by creating lookup table which includes the relative positions (offsets) of each letter in the two sequences. The highest offset frequency refers to the maximum possible number of matches which may be achieved. The offset refers to the number of shifts. For example, let sequences S and T are two unaligned sequences such that S: CACACT, T: GCACAC. Computing the offsets of each letter in the two sequences shows that the offset '-1' has the highest frequency which is '5', i.e., if the sequence S is shifted to the right by '1', this will create 5 matches.

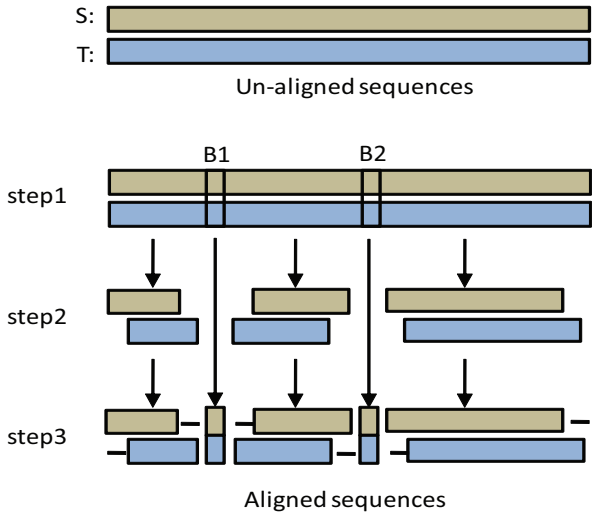

Fig. 1. Steps of our sequence alignment algorithm

The alignment score is computed as following:

$$\begin{aligned} \textit{Alignment Score} \quad = \quad & \textit{(\# of matches} \times \textit{match\_score)} \\ & + \textit{(\# of gaps} \times \textit{gap\_score)} \\ & + \textit{(\# of mismatches} \times \textit{mismatch\_score)} \end{aligned} \tag{1}$$

Usually, the match score has positive value. Therefore, more number of matches will increase the alignment score, contrary to the gap and the mismatch scores which have negative values to decrease the alignment score.

3- Insert the barriers between the aligned subsequences (step 3 in Fig. 1). The new alignment score is computed by adding

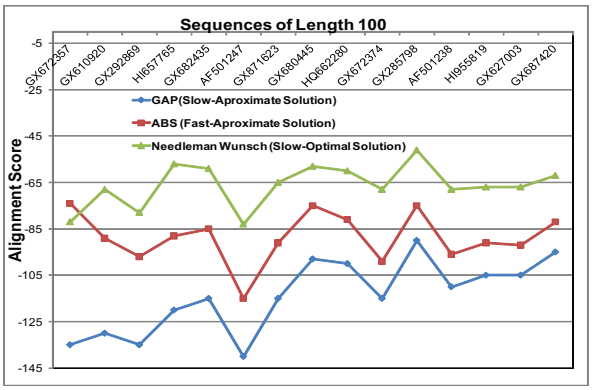

Fig. 2. Results of aligning sequences of length 100 using the GAP, the ABS and the Needleman-Wunsch algorithms

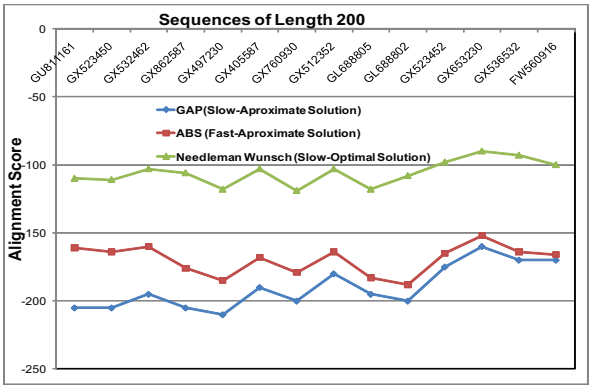

Fig. 3. Results of aligning sequences of length 200 using the GAP, the ABS and the Needleman-Wunsch algorithms

the matching score for each inserted barrier to the previously computed alignment score.

1. EXPERIMENTAL RESULTS

In this section, we present the experimental results of our ABS Algorithm in comparison to the approximate alignment algorithm "GAP", and the optimal alignment algorithm "Needleman-Wunsch". The evaluations are conducted using sequences of HUMAN division with length of 100, 200 and 500 nucleotides. The sequences are downloaded from the well known DNA sequences of the database "DNA Data Bank of Japan" (ddbj) [3]. In each figure, the y-axis shows the alignment score for the different programs and the x-axis shows the The accession number of the database sequences. The accession number of the queries are GX389453, HQ032039 and FQ377774, for the sequences of length 100, 200 and 500 Nucleotides, respectively.

Figures 2-4 shows that the alignment score of the ABS Algorithm is better than the score of the GAP Algorithm for all database sequences. On average, the percentage of improvement in the alignment score of the ABS versus the GAP is 51%, 22% and 28% for the length of 100, 200 and 500 nucleotides, respectively. Therefore, the ABS results are more close to the optimal results (Neddleman-Wunsch) when they are compared with the results of the GAP.

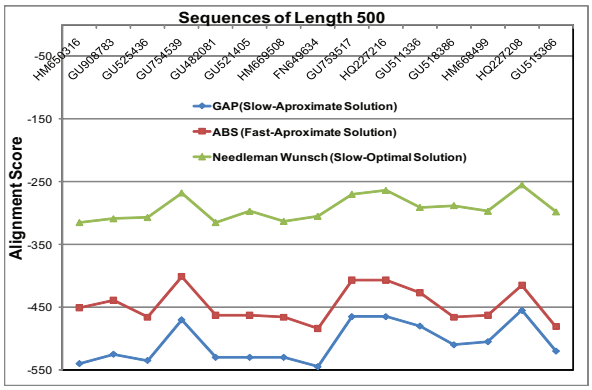

Fig. 4. Results of aligning sequences of length 500 using the GAP, the ABS and the Needleman-Wunsch algorithms

On the other hand, the ABS algorithm runs in time linear to the sequence length ($O(n)$) because it scans the two sequences together and splits them into many sub-sequences during the scan. In the case of the GAP or the Needleman-Wunsch, the algorithm runs in time proportional to the product of sequence lengths ($O(m \times n)$). Therefore, our ABS Algorithm is much faster than the GAP and the Needleman-Wunsch algorithms.

2. REFERENCES

[1] Needleman, S. and Wunsch, C. A general method applicable to the search for similarities in the amino acid sequence of two sequences. Journal of Molecular Biology, 48(3), 443-453, 1970

[2] M. Affan Zidan, Talal Bonny, and Khaled N. Salama. High Performance Technique for Database Applications Using a Hybrid GPU/CPU Platform. In the IEEE/ACM 21st Great Lake Symposium on VLSI. May 2011

[3] http://www.ddbj.nig.ac.jp/

[4] Xiaoqiu Huang1 and Kun-Mao Chao. A generalized global alignment algorithm. In Bioinformatics, 19, 228233. 2003.

[5] Huang X, Brutlag DL. Dynamic Use of Multiple Parameter Sets in Sequence Alignment. Nucleic Acids Research 2007, 35(2):678-686.

Number of words including references is 947