BIOINFORMATICS

Vol. 19 no. 2 2003
Pages 228–233

A generalized global alignment algorithm

Xiaoqiu Huang [1,*] and Kun-Mao Chao [2]

[1] Department of Computer Science, Iowa State University, 226 Atanasoff Hall, Ames, IA 50011-1040, USA and [2] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

Received on May 27, 2002; revised on August 28, 2002; accepted on September 2, 2002

ABSTRACT

Motivation: Homologous sequences are sometimes similar over some regions but different over other regions. Homologous sequences have a much lower global similarity if the different regions are much longer than the similar regions.

Results: We present a generalized global alignment algorithm for comparing sequences with intermittent similarities, an ordered list of similar regions separated by different regions. A generalized global alignment model is defined to handle sequences with intermittent similarities. A dynamic programming algorithm is designed to compute an optimal general alignment in time proportional to the product of sequence lengths and in space proportional to the sum of sequence lengths. The algorithm is implemented as a computer program named GAP3 (Global Alignment Program Version 3). The generalized global alignment model is validated by experimental results produced with GAP3 on both DNA and protein sequences. The GAP3 program extends the ability of standard global alignment programs to recognize homologous sequences of lower similarity.

Availability: The GAP3 program is freely available for academic use at http://bioinformatics.iastate.edu/aat/align/align.html.

Contact: xqhuang@cs.iastate.edu; kmchao@csie.ntu.edu.tw

INTRODUCTION

Global alignment algorithms are intended for comparing two sequences that are entirely similar (Needleman and Wunsch, 1970; Waterman *et al.*, 1976). Local alignment algorithms are intended for comparing sequences that contain locally similar regions (Smith and Waterman, 1981; Gotoh, 1982; Pearson and Lipman, 1988; Altschul *et al.*, 1990; Huang and Miller, 1991; Burkhardt *et al.*, 1999; Arslan *et al.*, 2001; Ma *et al.*, 2002). Those methods are very useful in analysis of DNA and protein sequences. In this paper, we generalize the global alignment algorithms to compare sequences with intermittent similarities, an ordered list of similar regions separated by different regions.

Homologous sequences are sometimes similar over some regions but different over other regions. For example, homologous genomic DNA sequences from related organisms such as human and mouse are usually similar over exon regions but different over intron regions. Homologous protein sequences are sometimes similar over some conserved domains but different over variable regions. Homologous sequences have a much lower global similarity if the different regions are much longer than the similar regions. We present a generalized global alignment model to address sequences with intermittent similarities and design a dynamic programming algorithm for computing an optimal general alignment of two sequences. The algorithm runs in time proportional to the product of sequence lengths and in space proportional to the sum of sequence lengths. The algorithm is implemented as a computer program named GAP3 (Global Alignment Program Version 3). The generalized global alignment model is validated by experimental results produced with the program on both DNA and protein sequences.

A number of fast comparison programs have been developed specially for comparing homologous and syntenic genomic DNA sequences (Delcher *et al.*, 1999; Jareborg *et al.*, 1999; Batzoglou *et al.*, 2000; Schwartz *et al.*, 2000). The GAP3 program is much slower than the fast comparison programs. For example, it takes GAP3 15 hours to compare two genomic sequences of 500 kb on an ordinary computer. Thus, it is not possible to compare two mammalian genomes with GAP3 on an ordinary computer. On the other hand, GAP3 is more sensitive than the fast comparison programs because GAP3 searches the entire solution space and produces an optimal solution. The improved sensitivity of optimal alignment algorithms over fast comparison algorithms is confirmed by Pearson (1995) in a comprehensive study. Thus, GAP3 is suitable for comparing protein sequences and short genomic sequences. For example, it takes GAP3 0.2 second to compare two protein se-

[*] To whom correspondence should be addressed.

quences of 1000 residues on an ordinary computer. The GAP3 program extends the ability of standard global alignment programs to recognize homologous protein sequences of lower similarity. Moreover, the generalized global alignment algorithm presented in this paper can be used as a basic pairwise alignment algorithm in a multiple sequence alignment program. For instance, the CLUSTAL W program (Thompson *et al.*, 1994) and the MAP program (Huang, 1994) are based on pairwise alignment algorithms.

ALIGNMENT MODEL

We define a generalized global alignment model to handle sequences with intermittent similarities. Let $A = a_1a_2 \ldots a_m$ and $B = b_1b_2 \ldots b_n$ be two sequences of lengths m and n. A generalized global alignment (or general alignment) of A and B consists of substitutions, gaps, and difference blocks. A substitution associates a residue of A with a residue of B. A gap consists only of residues from one sequence with each residue associated with the symbol '$-$'. There are two kinds of gaps. A deletion gap consists only of residues from A and an insertion gap consists only of residues from B. A difference block consists of residues from one or two sequences with each residue associated with the symbol '$+$'. There are three types of difference blocks. A difference block of type 1 consists only of residues from A, a difference block of type 2 consists only of residues from B, and a difference block of type 3 consists of residues from both A and B. As an example, a general alignment is shown in Figure 1. Let $\sigma(a, b)$ be the score of a substitution involving residues a and b. The score of a gap of length k is $-(q + k \times r)$, where nonnegative numbers q and r are gap-open and gap-extension penalties, respectively. The score of a difference block is $-d$, where nonnegative number d is a constant penalty for each difference block. The score of a general alignment is the sum of scores of each substitution, each gap, and each difference block in the alignment. An optimal general alignment is one with the maximum score.

An algorithm for computing an optimal general alignment of A and B is developed by dynamic programming. Let $A_i = a_1a_2 \ldots a_i$ and $B_j = b_1b_2 \ldots b_j$ be initial segments of lengths i and j of A and B. Define $S(i, j)$ to be the maximum score of general alignments of A_i and B_j. Then $S(m, n)$ is the score of an optimal general alignment of A and B. To compute the matrix S efficiently, three additional matrices are introduced. Define $H(i, j)$ to be the maximum score of general alignments of A_i and B_j that end with a difference block. Similarly, define $D(i, j)$ for general alignments that end with a deletion gap and $I(i, j)$ for general alignments that end with an insertion gap. The following recurrences for computing the matrices are derived from the definitions of the matrices:

```
GCGCTCCGGGACGCCTTCCGCCGTCGGGAGCCCTACAACTACCTGCAGAGGGCCTATTAC
+++++++++++++++++++++++++||||||| ||||||||||||||||||||||| |||
                         GGGAGCCTTACAACTACCTGCAGAGGGCCTACTAC

CAGGTGGGGAGCGGGCCGGGCAG                                  TAG
|||||| ||---||||||| |||+++++++++++++++++++++++++++++++++++++
CAGGTGCGG   GGGCCGGCCAGGGTGCTACCCCAAGCCTACTGACTGTCTTACTGG

CCTTCCCCAGAGCCCCCTAGCCGCAGGCACCAGAGGGTCCAAGACAAGACTGGAAGGGCA
+++++++++++++++++++++++|| ||  ||| | ||||| || ||  |||| | | |
                       CAAGCTTCAGCGAGTCCAGGAGAAAGCTGGGAAGCCC

CCTCGGGTTCGG      GAGGAGCTGTGAGTGGCT
|  ||||| |||------||||| |||||| |||||++++++++++++++++++++++++
CGCCGGGTCCGGGTCCGAGAGGAACTGTGAATGGCTGAGCCTGCTTCTCGAGGATCAGGC
```

Fig. 1. A general alignment of two DNA sequences. There are two gaps indicated by '$-$' and three difference blocks indicated by '$+$' on the alignment. Major differences between the sequences are represented by difference blocks, whereas minor differences are represented by mismatches and gaps.

$$
\begin{aligned}
S(0,0) &= 0,\\
S(i,0) &= \max\{D(i,0), H(i,0)\} \quad \text{for } i > 0,\\
S(0,j) &= \max\{I(0,j), H(0,j)\} \quad \text{for } j > 0,\\
S(i,j) &= \max\{S(i-1, j-1) + \sigma(a_i, b_j),\\
&\qquad D(i,j), I(i,j), H(i,j)\}\\
&\qquad \text{for } i > 0 \text{ and } j > 0.
\end{aligned}
$$

$$
\begin{aligned}
D(0,j) &= S(0,j) - q \quad \text{for } j \geq 0,\\
D(i,0) &= D(i-1,0) - r \quad \text{for } i > 0,\\
D(i,j) &= \max\{D(i-1,j) - r, S(i-1,j) - q - r\}\\
&\qquad \text{for } i > 0 \text{ and } j > 0.
\end{aligned}
$$

$$
\begin{aligned}
I(i,0) &= S(i,0) - q \quad \text{for } i \geq 0,\\
I(0,j) &= I(0,j-1) - r \quad \text{for } j > 0,\\
I(i,j) &= \max\{I(i,j-1) - r, S(i,j-1) - q - r\}\\
&\qquad \text{for } i > 0 \text{ and } j > 0.
\end{aligned}
$$

$$
\begin{aligned}
H(i,j) &= -d \quad \text{for } i = 0 \text{ or } j = 0,\\
H(i,j) &= \max\{H(i,j-1), H(i-1,j),\\
&\qquad S(i,j-1) - d, S(i-1,j) - d\}\\
&\qquad \text{for } i > 0 \text{ and } j > 0.
\end{aligned}
$$

The matrix H is the major difference between the generalized alignment model and the standard global alignment model. Figure 2 illustrates how an entry in the matrix H is computed. Below we justify the recurrence for the matrix H. For $i > 0$ and $j > 0$, partition into four groups the general alignments of A_i and B_j ending with a difference block. A general alignment is in group 1 if the last difference block consists only of a residue at position i of A. A general alignment is in group 2 if the last difference block consists of a residue

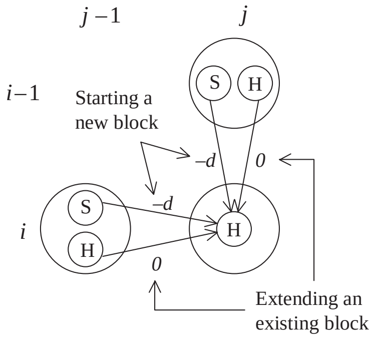

Fig. 2. Four cases for computing $H(i, j)$.

at position i of A and other residues. Groups 3 and 4 are similarly defined with respect to a residue at position j of B. Note that the last difference block must contain a residue at position i of A or position j of B. Let $P_1(A_i, B_j)$ be a largest-scoring alignment in group 1. Note that $P_1(A_i, B_j)$ ends with a difference block consisting only of a residue a at position i of A. Let $Q_1(A_{i-1}, B_j)$ denote the entire portion of $P_1(A_i, B_j)$ before the residue a. Since the difference block consists only of the residue a, the alignment $Q_1(A_{i-1}, B_j)$ ends with a substitution or gap and its score has to be $S(i-1, j)$. Thus, the score of the alignment $P_1(A_i, B_j)$ is equal to $S(i-1, j) - d$, the score of the alignment $Q_1(A_{i-1}, B_j)$ minus the penalty for the difference block. Because $P_1(A_i, B_j)$ is a largest-scoring alignment in group 1, the maximum score of alignments in group 1 is $S(i-1, j) - d$.

Next we consider group 2. Let $P_2(A_i, B_j)$ be a largest-scoring alignment in group 2. Let $Q_2(A_{i-1}, B_j)$ denote the entire portion of $P_2(A_i, B_j)$ before a residue a at position i of A. Since the difference block consists of the residue a and other residues, the alignment $Q_2(A_{i-1}, B_j)$ ends with a difference block and its score has to be $H(i-1, j)$. The score of the alignment $P_2(A_i, B_j)$ is equal to $H(i-1, j)$ because the penalty for the difference block is already included in $H(i-1, j)$. Thus, the maximum score of alignments in group 2 is $H(i-1, j)$. Similarly, we can show that the maximum score of alignments in group 3 is $S(i, j-1) - d$ and that the maximum score of alignments in group 4 is $H(i, j-1)$. Thus, for $i > 0$ and $j > 0$, $H(i, j)$ is equal to the maximum of the four expressions, each being the maximum score of alignments in a group. Assume that A_0 and B_0 denote the empty sequence of length 0. For $i = 0$ or $j = 0$, a general alignment of A_i and B_j ending with a difference block can not contain any match. A largest-scoring general alignment has to be an alignment consisting only of one difference block and its score is $-d$. Thus, for $i = 0$ or $j = 0$, $H(i, j) = -d$.

The matrices are computed according to the recurrences in order of rows. The computation is performed by saving only the most recent row of each matrix. This is possible because each of the matrices observes the property that the score at an entry depends only on scores at its neighbor entries. An optimal general alignment is computed in linear space by an algorithm described in the next section.

What value should be used for the parameter d? As shown in Figure 3, a general alignment is simply an ordered list of local alignments separated by difference blocks, where a local alignment consists only of substitutions and gaps. Consider an optimal general alignment T consisting of two or more difference blocks. Let t be an internal local alignment of T, which is between two difference blocks. Let T' be the general alignment obtained from T by replacing the local alignment t and the two difference blocks by one difference block. Then we have

$$score(T) = score(T') + score(t) - d.$$

Because T is optimal, we have $score(T) \geq score(T')$ and $score(t) \geq d$. We conclude that for any optimal general alignment with two or more difference blocks, the score of each internal local alignment is greater than or equal to d. Thus, the parameter d should be set to the minimum score of desirable local alignments.

ALGORITHM

We develop a space-efficient algorithm for computing an optimal general alignment of A and B. A divide-conquer technique is developed by Hirschberg (1975) for computing a longest common subsequence of two sequences. The Hirschberg technique is applied to global alignment models (Myers and Miller, 1988; Huang, 1994; Mott, 1997). Here the Hirschberg technique is applied to the generalized alignment model. The main idea of the space-efficient algorithm is to determine a middle pair of positions on an optimal general alignment in linear space. Then the portions of the optimal general alignment before and after the middle pair of positions are constructed recursively. Let $imid$ be $\lfloor m/2 \rfloor$, where $\lfloor y \rfloor$ is the largest integer less than or equal to y. We first consider a procedure for finding a position $jmid$ such that the pair of positions $imid$ and $jmid$ is on an optimal general alignment of A and B.

Partition the general alignments of A and B into three groups. Group 1 consists of general alignments with a difference block containing a residue a at position $imid$ of A and another residue immediately to the right of a. In other words, a general alignment is in group 1 if, upon splitting the alignment into two parts immediately after the residue a, the first part ends with a difference block and the second part begins with a difference block. Group 2 consists of general alignments with a deletion gap

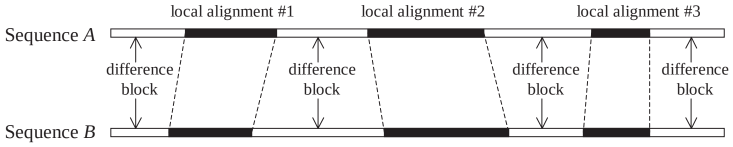

Fig. 3. A general global alignment is an ordered list of local alignments separated by difference blocks.

containing residues at positions $imid$ and $imid + 1$ of A. Group 3 consists of all the remaining general alignments. Note that a residue at position $imid$ of A can not be inside any insertion gap and hence that there is no need to consider this case. We consider computing the score of and a middle pair of positions on a largest-scoring alignment in each group.

Let $R(A, B)$ denote a largest-scoring alignment of A and B in group 1. Split $R(A, B)$ into two parts immediately after position $imid$ of A. Let jh be the largest position of B in the first part. Let A_i^s denote the suffix $a_{i+1}a_{i+2}\ldots a_m$ of A. Notation B_j^s is similarly defined. Then the first part of $R(A, B)$ is an alignment, denoted by $R_1(A_{imid}, B_{jh})$, of A_{imid} and B_{jh} and the second part is an alignment, denoted by $R_2(A_{imid}^s, B_{jh}^s)$, of A_{imid}^s and B_{jh}^s. Note that $R_1(A_{imid}, B_{jh})$ ends with a difference block and that $R_2(A_{imid}^s, B_{jh}^s)$ begins with a difference block. Define $\bar{H}(i, j)$ to be the maximum score of general alignments of A_i^s and B_j^s that begin with a difference block. It follows from the definition of $R(A, B)$ that the score of $R_1(A_{imid}, B_{jh})$ is $H(imid, jh)$ and the score of $R_2(A_{imid}^s, B_{jh}^s)$ is $\bar{H}(imid, jh)$. Moreover, the score of $R(A, B)$, denoted by hk, is

$$hk = H(imid, jh) + \bar{H}(imid, jh) + d,$$

where including the term d on the right-hand side ensures that the difference block containing a residue at position $imid$ of A is charged for a penalty exactly once. Observe that for each j, $0 \le j \le n$, $H(imid, j) + \bar{H}(imid, j) + d$ is the score of an alignment of A and B in group 1. Combining the observations together, we obtain

$$hk = \max\{H(imid, j) + \bar{H}(imid, j) + d \mid 0 \le j \le n\}.$$

Note that jh is a position at which the maximum score hk is obtained. Thus, the score of and a middle pair of positions on a largest-scoring alignment in group 1 can be obtained using middle rows of the matrices H and $\bar{H}$.

The score of and a middle pair of positions on a largest-scoring alignment in group 2 and those in group 3 can be obtained similarly. Define $\bar{D}(i, j)$ to be the maximum score of general alignments of A_i^s and B_j^s that begin with a deletion gap. Define $\bar{S}(i, j)$ to be the maximum score of general alignments of A_i^s and B_j^s. Then the score of a largest-scoring alignment in group 2, denoted by df, is

$$df = \max\{D(imid, j) + \bar{D}(imid, j) + q \mid 0 \le j \le n\}.$$

Let jd be a position at which the maximum score df is obtained. Then $(imid, jd)$ is a middle pair of positions on a largest-scoring alignment in group 2. In group 3, the score of a largest-scoring alignment, denoted by st, is

$$st = \max\{S(imid, j) + \bar{S}(imid, j) \mid 0 \le j \le n\}.$$

Let js be a position at which the maximum score st is obtained. Then $(imid, js)$ is a middle pair of positions on a largest-scoring alignment in group 3. The recurrences for computing the matrices $\bar{D}$, $\bar{I}$, $\bar{H}$, and $\bar{S}$ are developed in the same way as those for D, I, H, and S. The score of an optimal general alignment of A and B is $\max\{df, hk, st\}$. Let $jmid$ be the corresponding one of jd, jh, and js. Then the pair of positions $imid$ and $jmid$ is on an optimal general alignment of A and B.

An algorithm for computing an optimal general alignment of A and B in linear space consists of the following steps. If m is small enough, compute an optimal general alignment of A and B using a traceback procedure. Otherwise, determine a pair of positions $imid$ and $jmid$ on an optimal general alignment of A and B, and recursively compute the portions of the alignment before and after the pair of positions.

The positions $imid$ and $jmid$ are determined as follows. Set $imid = \lfloor m/2 \rfloor$. Compute the matrices D, I, H, and S from row 0 to row $imid$, and save $D(imid, j)$, $H(imid, j)$, and $S(imid, j)$ for $0 \le j \le n$. Compute the matrices $\bar{D}$, $\bar{I}$, $\bar{H}$, and $\bar{S}$ from row m down to row $imid$, and save $\bar{D}(imid, j)$, $\bar{H}(imid, j)$, and $\bar{S}(imid, j)$ for $0 \le j \le n$. Compute the values df, hk and st. Let jd be a position at which the maximum score df is obtained, jh a position at which the maximum score hk is obtained, and js a position at which the maximum score st is obtained. If $df > hk$ and $df > st$, then set $jmid = jd$. Otherwise, if $hk > df$ and $hk > st$, then set $jmid = jh$. Otherwise, set $jmid = js$.

It can be proved that the algorithm requires memory and time in proportion to the sum and product of sequence lengths, respectively. The proof is similar to one in Huang (2002).

RESULTS

The new algorithm is implemented as a computer program named GAP3. The GAP3 program can handle both DNA and protein sequences. The program takes as input two sequences in FASTA format. The parameters for the program are substitution score table σ, gap open penalty q, gap extension penalty r, and difference block penalty d. On DNA sequences, $\sigma(a, a) = 10$ for each base a and $\sigma(a, b) = ms$ for $a \neq b$, where ms is a mismatch score parameter. On protein sequences, σ is a PAM or BLOSUM score table specially formatted for GAP3. We tested GAP3 on DNA and protein sequences. The results indicate that the new algorithm in GAP3 almost worked as expected.

The only unexpected feature of the algorithm was that an optimal general alignment produced by GAP3 occasionally begins with an isolated match, which is followed by a difference block, or ends with an isolated match, which is preceded by a difference block. Although the alignment is mathematically optimal, the isolated match in the beginning or end of the alignment is not biologically meaningful. The alignment model presented in Section 2 was modified such that any general alignment beginning or ending with a local alignment of score less than d is not optimal. The score of a general alignment is revised as follows. If a general alignment does not begin with a difference block, an extra penalty of d is subtracted from the score of the general alignment. Similarly, if a general alignment does not end with a difference block, an extra penalty of d is subtracted from the score of the general alignment. The recurrences for the matrix S is modified accordingly. The value $S(0, 0)$ is set to $-d$ instead of 0, and the score of an optimal general alignment is $\max\{S(m, n) - d, H(m, n)\}$ instead of $S(m, n)$. The results presented below were obtained with modified GAP3.

An alternative solution to the problem mentioned above is to charge no penalty for initial and terminal difference blocks. The solution involves making the following modifications. The value 0 is included in the recurrences for the matrix S for $i \geq 0$ and $j \geq 0$. The score of an optimal general alignment is $\max\{S(m, n), H(m, n) + d\}$.

The GAP3 program was used to compare two syntenic human and mouse sequences containing 17 genes. The human sequence is of 222 930 bp (GenBank Accession U47924) and the mouse sequence is of 227 538 bp (GenBank Accession AC002397). A value for the parameter d was selected based on internal exon lengths. Internal exons are often of length at least 50 bp. The score of 50 matches at 10 per match is 500. A value of 300 was used for the parameter d. Values for the other parameters were chosen based on our prior experiences with standard alignment programs: $ms = -20$, $q = 60$, and $r = 2$. The human and mouse sequences were screened for repeats with RepeatMasker (Smit and Green, 1997). The masked sequences were used as input to GAP3. GAP3 produced a general alignment of the two sequences in 2.5 hours on an entry-level Linux PC. The alignment consists of 154 similar regions separated by difference blocks. The alignment fully displays the similar regions but omits most of the difference blocks. The 154 similar regions are mostly coding exon regions and untranslated regions. Gaps occur much more frequently in alignments of untranslated regions than in alignments of coding exon regions. The total length of the 154 similar regions is 43,445 bp and their average identity is 79%. The 154 similar regions constitute about 19% of each of the two sequences. A portion of the alignment is available at the URL address for downloading GAP3.

The GAP3 program was used to compute an optimal general alignment of two protein sequences from an immunoglobulin A protease family (Pfam accession no. PF02395). One is an *H. influenzae* protein of 1409 residues (Swiss–Prot accession no. P44596) and the other an *E. coli* protein of 1306 residues (TrEMBL accession no. P77070). The two sequences have a global identity of 18%. The following values were used for the parameters: σ is BLOSUM62, $q = 10$, $r = 2$, and $d = 40$. GAP3 produced a general alignment of the two sequences in 0.3 second on an entry-level Linux PC. The alignment contains six similar regions with a total length of 581 residues and an average identity of 30%. The alignment is available at the URL address for downloading GAP3. The percent identity of 30% over the length of 581 residues is a stronger evidence for indicating that the two protein sequences are homologous. The 6 similar regions constitute 41% of the longer protein sequence and 44% of the shorter protein sequence.

DISCUSSION

We have generalized standard dynamic programming algorithms based on Needleman and Wunsch (1970) to compare sequences with intermittent similarities. The proposed method complements fast existing methods for comparing syntenic genomic DNA sequences. The fast existing methods work well on sequences with highly similar regions because they perform searches in much smaller space indicated by similar regions. The proposed method is much slower but is able to find lower similarities between sequences because it explores the entire search space and computes an optimal solution. The proposed method is suitable for comparing short genomic regions with lower similarities. The proposed method can serve as a general pairwise alignment program for a multiple alignment program on protein sequences, which produces initial protein sequence alignments for HMM-based methods such as HMMER (Eddy, 1998).

Although the proposed algorithm produces a list of local similarities consistent in order, the method is intended for finding a global similarity of two sequences. To find local similarities that are order-independent, algorithms based on Smith and Waterman (1981) and their fast variations should be used. Note that local alignment algorithms can also be used in a two-step method to find a list of local similarities consistent in order. In the two-step method, all significant local similarities are first computed and then an ordered list of local similarities with the maximum sum of scores is generated. The two-step method is likely to produce identical results as the proposed algorithm in practice. However, the two-step method is not guaranteed to produce an optimal alignment in theory because the computation and ordering of local similarities are performed separately. The two-step method may take more time than the proposed algorithm, which always takes quadratic time.

ACKNOWLEDGEMENTS

We thank Volker Brendel, Duane Enger and Shannon Schlueter for suggestions. X. Huang was supported in part by NIH Grants R01 HG01502-05 and R01 HG01676-05 from NHGRI. K.-M. Chao was supported in part by grant NSC 90-2213-E-010-003 from the National Science Council, Taiwan.

REFERENCES

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Arslan,A., Eğecioğlu,Ö and Pevzner,P. (2001) A new approach to sequence comparison: normalized sequence alignment. *Bioinformatics*, **17**, 327–337.

Batzoglou,S., Pachter,L., Mesirov,J.P., Berger,B. and Lander,E.S. (2000) Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res.*, **10**, 950–958.

Burkhardt,S., Crauser,A., Lenhof,H.-P., Rivals,E., Ferragina,P. and Vingron,M. (1999) Q-gram based database searching using a suffix array. *In Third Annual International Conference on Computational Molecular Biology*. pp. 11–14.

Delcher,A.L., Kasif,S., Fleischmann,R.D., Peterson,J., While,O. and Salzberg,S.L. (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Hirschberg,D.S. (1975) A linear space algorithm for computing maximal common subsequences. *Commun. Assoc. Comput. Mach.*, **18**, 341–343.

Huang,X. (1994) On global sequence alignment. *Comput. Appl. Biosci.*, **10**, 227–235.

Huang,X. (2002) Bio-sequence comparison and applications. In Jiang,T., Xu,Y. and Zhang,M. (eds), *Current Topics in Computational Molecular Biology*. MIT Press, Cambridge, pp. 45–69.

Huang,X. and Miller,W. (1991) A time-efficient, linear-space local similarity algorithm. *Adv. Appl. Math.*, **12**, 337–357.

Jareborg,N., Birney,E. and Durbin,R. (1999) Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Res.*, **9**, 815–824.

Ma,B., Tromp,J. and Li,M. (2002) PatternHunter: faster and more sensitive homology search. *Bioinformatics*, **18**, 440–445.

Mott,R. (1997) EST_GENOME: a program to align spliced DNA sequences to unspliced genomic DNA. *Comput. Appl. Biosci.*, **13**, 477–478.

Myers,E.W. and Miller,W. (1988) Optimal alignments in linear space. *Comput. Appl. Biosci.*, **4**, 11–17.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Pearson,W.R. (1995) Comparison of methods for searching protein sequence databases. *Protein Sci.*, **4**, 1145–1160.

Pearson,W.R. and Lipman,D. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Schwartz,S., Zhang,Z., Frazer,K., Smit,A., Riemer,C., Bouck,J., Gibbs,R., Hardison,R. and Miller,W. (2000) PipMaker–a web server for aligning two genomic DNA sequences. *Genome Res.*, **10**, 577–586.

Smit,A. and Green,P. (1997) http://ftp.genome.washington.edu/RM/RepeatMasker.html.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Waterman,M.S., Smith,T.F. and Beyer,W.A. (1976) Some biological sequence metrics. *Adv. Math.*, **20**, 367–387.