J. Mol. Biol. (1982) **162**, 705–708

An Improved Algorithm for Matching Biological Sequences

The algorithm of Waterman *et al.* (1976) for matching biological sequences was modified under some limitations to be accomplished in essentially MN steps, instead of the M^2N steps necessary in the original algorithm. The limitations do not seriously reduce the generality of the original method, and the present method is available for most practical uses. The algorithm can be executed on a small computer with a limited capacity of core memory.

The currently used major algorithms for aligning biological sequences (protein and nucleic acid sequences) stem from the pioneering work of Needleman & Wunsch (1970). Needleman–Wunsch's method has also been applied to statistical tests of relatedness between a pair of sequences (Barker & Dayhoff, 1972; Doolittle, 1981). Sellers (1974) proved that evolutionary distances obtained with a similar algorithm to that of Needleman & Wunsch satisfy metric conditions. Sellers' metric was later generalized by Waterman *et al.* (1976) so that deletions/insertions (gaps) of any length are allowed. Inclusion of multiple-sized gaps is feasible for comparing biological sequences since a long gap can be produced by a single mutational event. This situation is incorporated into the method of Waterman *et al.* (1976) by assigning a weight $w_k \leq kw_1$ to a gap of length k, whereas the gap weight is confined to $w_k = kw_1$ for all k values in the method of Needleman, Wunsch and Sellers. However, the algorithm of Waterman *et al.* (1976) has a drawback in that it takes a large number of computational steps of the order of M^2N compared to the MN steps of Needleman–Wunsch–Sellers' algorithm, where M and N ($M \geq N$) are the lengths of the proteins or nucleic acids under comparison. This is a particularly serious problem when calculations are made on a low-speed small computer.

In this letter I present a new algorithm which allows multiple-sized gaps but runs in essentially MN steps if the gap weight has a special form of $w_k = uk + v$ ($u \geq 0$, $v \geq 0$). This form of gap weight has usually been used in computer systems that adopt the matching algorithm of Waterman *et al.* (Smith *et al.*, 1981; Kanehisa, 1982).

(a) *Algorithm*

Let the two sequences be $A = a_1 a_2 \ldots a_M$ and $B = b_1 b_2 \ldots b_N$. A weight $d(a_m, b_n)$ is given to an aligned pair of residues a_m and b_n. Usually, $d(a_m, b_n) \leq 0$ if $a_m = b_n$, and $d(a_m, b_n) > 0$ if $a_m \neq b_n$. The algorithm of Waterman *et al.* (1976) generates a distance matrix $D_{m,n}$ by an induction as follows:

$$D_{m,n} = \mathrm{Min}\,[D_{m-1,n-1} + d(a_m, b_n),\ P_{m,n},\ Q_{m,n}], \tag{1}$$

where

$$P_{m,n} = \underset{1 \leq k \leq m}{\mathrm{Min}}\,[D_{m-k,n} + w_k] \tag{2}$$

0022–2836/82/350705–04 $03.00/0 © 1982 Academic Press Inc. (London) Ltd.

and

$$Q_{m,n} = \underset{1 \le k \le n}{\text{Min}} \, [D_{m,n-k} + w_k]. \tag{3}$$

Although $P_{m,n}$ (or $Q_{m,n}$) appears to be calculated in $m-1$ (or $n-1$) steps, it can be obtained in a single step according to the following recursion relations:

$$\begin{aligned} P_{m,n} &= \text{Min}\,[D_{m-1,n} + w_1, \underset{2 \le k \le m}{\text{Min}} \, (D_{m-k,n} + w_k)] \\ &= \text{Min}\,[D_{m-1,n} + w_1, \underset{1 \le k \le m-1}{\text{Min}} \, (D_{m-1-k,n} + w_{k+1})] \\ &= \text{Min}\,[D_{m-1,n} + w_1, \underset{1 \le k \le m-1}{\text{Min}} \, (D_{m-1-k,n} + w_k) + u] \\ &= \text{Min}\,[D_{m-1,n} + w_1, P_{m-1,n} + u] \end{aligned} \tag{4}$$

and

$$Q_{m,n} = \text{Min}\,[D_{m,n-1} + w_1, Q_{m,n-1} + u]. \tag{5}$$

Thus, the induction is completed in MN steps, each of which consists of choosing the smallest of three numbers for $D_{m,n}$, and the smaller of two numbers for $P_{m,n}$ or $Q_{m,n}$.

At the beginning of the induction, one may set $D_{m,0} = P_{m,0} = w_m$ $(1 \le m \le M)$, and $D_{0,n} = Q_{0,n} = w_n$ $(1 \le n \le N)$. Alternatively, $D_{m,0} = P_{m,0} = 0$ and $D_{0,n} = Q_{0,n} = w_n$, or $D_{m,0} = P_{m,0} = 0$ and $D_{0,n} = Q_{0,n} = 0$ may be chosen in searching for the most locally similar subsequence (Sellers, 1980; Smith & Waterman, 1981; Goad & Kanehisa, 1982).

In a computer program, not all the elements of $D_{m,n}$, $P_{m,n}$ and $Q_{m,n}$ need be memorized; two one-dimensional arrays and one variable are sufficient to store temporary values of these quantities. This feature is also useful for executing the algorithm on a small computer equipped with a small size of core memory.

The optimally matched alignments are available by backtracking guided by the "direction matrix" $e_{m,n}$, whose element is a three-bit binary number indicating the paths through which the minimum value of $D_{m,n}$ is chosen (Smith *et al.*, 1981; Goad & Kanehisa, 1982). The complete set of $e_{m,n}$ values is obtained by running the above algorithm twice, first calculating $e_{m-k,n}$ $(k \ge 0)$ and second $e_{n-k,m}$ exchanging the column/row assignments of A and B, and finally taking bit-to-bit logical OR values of the first $e_{m,n}$ and the second $e_{n,m}$.

Figure 1 shows an example, in which $e_{m,n}$ values obtained after the first run of the algorithm are shown in (a), and the final $e_{m,n}$ values in (b). Figure 1(a) and (b) also demonstrates $D_{m,n}$ values and $Q_{m,n}$ values, respectively. Note that the second run converts the underlined $e_{m,n}$ values from one to five, although they do not contribute to the traceback (indicated by arrows) in this example.

The above-mentioned algorithm can be further generalized if w_k has the following form: $w_k = u_0 k + v$ $(1 \le k \le K_1)$, $w_k = u_1(k - K_1) + w_{K_1}$ $(K_1 < k \le K_2)$, ..., $w_k = u_L(k - K_L) + w_{K_L}$ $(K_L < k)$, where u_l terms are constants of $u_0 > u_1 > u_2 \ldots > u_L \ge 0$. The simplest case of interest is $L = 1$ and $u_1 = 0$, i.e. w_k is a linear function of k in the range $1 \le k \le K_1$, while it is a constant $(= w_{K_1})$ for all k values greater than

	Δ	A	A	A	T	T
Δ	0	12	22	32	42	52
A	12	0	12	22	42	52
		1	*3*	*3*	*3*	*3*
A	22	12	0	12	32	42
		1	*1*	*3*	*3*	*2*
A	32	22	12	0	22	32
		1	*1*	*1*	*3*	*2*
G	42	42	32	22	10	32
		1	*1*	*5*	*1*	*3*
G	52	52	42	32	32	20
		1	*4*	*4*	*5*	*1*
T	62	62	52	42	32	32
		1	*4*	*4*	*1*	*1*
T	72	72	62	52	42	32
		1	*4*	*4*	*1*	*1*

(a)

	Δ	A	A	A	T	T
Δ	0	12	22	32	42	52
A	12	34	44	54	64	74
		1	*3*	*3*	*3*	*3*
A	22	22	34	44	64	74
		5	*1*	*3*	*3*	*2*
A	32	32	22	34	54	64
		5	*5*	*1*	*3*	*2*
G	42	42	32	22	44	54
		5	*5*	*5*	*1*	*3*
G	52	52	42	32	32	54
		5	*4*	*4*	*5*	*1*
T	62	62	52	42	42	44
		5	*4*	*4*	*1*	*1*
T	72	72	62	52	52	64
		5	*4*	*4*	*1*	*1*

(b)

FIG. 1. An example of operation of the algorithm. (a) $D_{m,n}$ (Arabic), and $e_{m,n}$ (Italic) obtained after the first run. (b) $Q_{m,n}$ (Arabic), and the completed $e_{m,n}$ (Italic). The underlined $e_{m,n}$ values are altered by the second run. The arrows indicate the paths of backtracking. To avoid going the wrong way, such as in the way shown by broken arrows, we always go straight ahead, if possible, at each branch point. The weight values used are $d(\mathrm{a}_m, \mathrm{b}_n)=0$ if $\mathrm{a}_m=\mathrm{b}_n$, $d(\mathrm{a}_m, \mathrm{b}_n)=10$ if $\mathrm{a}_m \neq \mathrm{b}_n$, and $w_k=10k+12$.

K_1. Such an assignment of w_k seems adequate for alignment of sequences with large gaps, e.g. alignment of *Halococcus morrhuae* 5 S RNA (Luehrsen *et al.*, 1981) against usual prokaryotic or eukaryotic 5 S RNAs. When $L=1$, the recursion relations for $P_{m,n}$ are derived as:

$$P^0_{m,n}=\mathrm{Min}\ [D_{m-1,n}+w_1,\ P^0_{m-1,n}+u_0], \tag{6}$$

$$P^1_{m,n}=\mathrm{Min}\ [D_{m-K_1-1,n}+w_{K_1}+u_1,\ P^1_{m-1,n}+u_1] \tag{7}$$

and

$$P_{m,n}=\mathrm{Min}\ [P^0_{m,n},\ P^1_{m,n}]. \tag{8}$$

The relations for $Q_{m,n}$ are obtainable analogously. These relations are also easily extended for $L \geq 2$.

To execute the above procedure on a computer, one needs to prepare a queue memory with $M \times (K_L+1)$ cells storing $D_{m,n-k}$ $(0 \leq k \leq K_L)$ values, in addition to $(L+1)$ one-dimensional arrays and $L+1$ variables which store temporary values of $P^l_{m,n}$ and $Q^l_{m,n}$ $(l=0,1,2,\ldots,L)$. The number of computational steps is roughly proportional to $(L+2)MN$, if $N \gg K_L$.

We cannot *a priori* determine appropriate values for the weights and parameters involved in the algorithm, but they may be estimated by a dynamic optimization procedure (Sankoff *et al.*, 1976). The weights thus obtained are useful for examining previously unknown relatedness between a pair of sequences. Such an investigation on the interrelation of 4·5 S RNA sequences is reported elsewhere (Takeishi & Gotoh, 1982).

I thank Dr M. I. Kanehisa for sending me the complete source codes of the Los Alamos

Sequence Analysis Package. I also thank Dr K. Takeishi and Dr Y. Tagashira for discussion and encouragement.

Department of Biochemistry
Saitama Cancer Center Research Institute
Ina-machi, Saitama 362, Japan

OSAMU GOTOH

Received 28 June 1982

REFERENCES

Barker, W. C. & Dayhoff, M. O. (1972). In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.), vol. 5, pp. 101–110, National Biomedical Research Foundation, Silver Spring.

Doolittle, R. F. (1981). *Science*, **214**, 149–159.

Goad, W. B. & Kanehisa, M. I. (1982). *Nucl. Acids Res.* **10**, 247–263.

Kanehisa, M. I. (1982). *Nucl. Acids Res.* **10**, 183–196.

Luehrsen, K. R., Nicholson, D. E., Eubanks, D. C. & Fox, G. E. (1981). *Nature (London)*, **293**, 755–756.

Needleman, S. B. & Wunsch, C. D. (1970). *J. Mol. Biol.* **48**, 443–453.

Sankoff, D., Cedergren, R. J. & Lapalme, G. (1976). *J. Mol. Evol.* **7**, 133–149.

Sellers, P. H. (1974). *J. Appl. Math. (Siam)*, **26**, 787–793.

Sellers, P. H. (1980). *J. Algorithm*, **1**, 359–373.

Smith, T. F. & Waterman, M. S. (1981). *J. Mol. Biol.* **147**, 195–197.

Smith, T. F., Waterman, M. S. & Fitch, W. M. (1981). *J. Mol. Evol.* **18**, 38–46.

Takeishi, K. & Gotoh, O. (1982). *J. Biochem. (Tokyo)*, **92**, 1173–1177.

Waterman, M. S., Smith, T. F. & Beyer, W. A. (1976). *Advan. Math.* **20**, 367–387.

Edited by S. Brenner